

Дискретна математика

Exonaut

13 май 2021 г.

Съдържание

1	Лекция 1: Логика и логически оператори	6
1.1	Дефиниции	6
1.2	Логически оператори	7
1.2.1	Отрицание(NOT)	7
1.2.2	И, Конюнкция (AND)	7
1.2.3	ИЛИ, Дизюнкция (OR)	7
1.2.4	Сума по модул 2, изключващо или (XOR)	8
1.2.5	Импликация, следствие	8
1.2.6	Двупосочно следствие	8
1.3	Закони за еквивалентни преобразувания	9
1.4	Предикатни функции и предикати	10
1.5	Квантор	10
1.6	Закони на Де Морган за квантори	10
2	Лекция 2: Математически доказателства	11
2.1	Теория на доказателствата	11
2.2	Терминология	11
2.3	Правила за извод	12
2.4	Аргументи	13
2.5	Правила за извод при използване на квантори и предикати	13
2.6	Доказателство на теореми	14
3	Лекция 3: Теория на множества	15
3.1	Парадокс на Ръсел	15
3.2	Множества	16
3.2.1	Равенство на множества	16
3.2.2	Подмножества	16
3.2.3	Собствени подмножества	17
3.2.4	Число на кардиналност, мощност(брой елементи) на множество	17
3.2.5	Множество от всички подмножества на дадено мно- жество	17
3.2.6	Декартово произведение	18
3.3	Операции на множества	18
3.3.1	Обединение	18
3.3.2	Сечение	18

3.3.3	Разлика	18
3.3.4	Допълнение	19
3.4	Идентичност на законите за логическо преобразуване и законите за операции с множества	19
4	Лекция 4: Релации (Отношения)	21
4.1	Релации	21
4.2	Функциите като релации	21
4.3	Релации, дефинирани върху множество	21
4.4	Свойства на релациите	22
4.5	Операции с релациите	23
4.5.1	Комбиниране на релациите	23
4.6	n-арни релации	24
4.7	Бази данни и релации	24
4.8	Представяне релации	25
4.8.1	Булеви матрици	25
4.8.2	Ориентирани графи	27
4.9	Релации на еквивалентност	27
4.10	Класове на еквивалентност	28
5	Лекция 5: Функции	30
5.1	Функциите като релации	31
5.2	Свойства на функции	31
5.2.1	Инекция	31
5.2.2	Строго нарастваща/намаляваща	32
5.2.3	Сюрекция	32
5.2.4	Биекция	32
5.3	Обратна функция	32
5.4	Крайни и безкрайни множества	32
5.5	Принцип на Дирихле	33
5.6	Композиция	33
5.7	Функции най-голямо цяло (Floor) и най-малко цяло (Ceiling)	34
6	Лекция 6: Булева алгебра	35
6.1	Булеви операции	35
6.2	Булеви (двоични) функции и изрази	35
6.3	Дуалност	37
6.4	Дефиниция на булевата алгебра	37

6.5	Логика, множества и булева алгебра	38
7	Лекция 7: Графи	39
7.1	Дефиниции и терминология	39
7.2	Специални графи	41
7.3	Представяне на графите	42
7.4	Изоморфизъм на графи	43
7.5	Пътища и контури в графи	43
7.6	Свързаност	44
7.7	Достижимост	45
7.8	Графи с допълнителни характеристики на ребрата	46
7.9	Най-къс път	46
7.10	Задача на търговския пътник	46
8	Лекция 8: Дървета	47
8.1	Дървета	47
8.2	Дървета с корен и наредба на върховете	48
8.3	Свойства на дърветата	48
8.4	Покриващо дърво	49
8.5	Минимално покриващо дърво (МПД)	49
8.6	Метод на Крускал	50
8.7	Метод на Прим	50
8.8	Особености на методите на Крускал и Прим	51
9	Лекция 9: Комбинаторика	52
9.1	Принцип на Дирихле	52
9.2	Принцип на добавянето	52
9.3	Принцип на умножението	53
9.4	Принцип на включването и изключването	53
9.5	Дървовидни структури за изброяване	54
9.6	Пермутации и комбинации	54
9.6.1	Пермутации	54
9.6.2	Комбинации	55
10	Лекция 9А: Теория на вероятностите	56
10.1	Опит, събитие, вероятност	56
10.2	Вероятност на противоположно събитие	59
10.3	Вероятност на обединение от събития	59

10.4	Теория на вероятностите	60
10.4.1	Равномерно разпределениена вероятностите	60
10.4.2	Вероятност на събитие	61
10.4.3	Обединение на несъвместими събития	61
10.5	Условна вероятност	62
10.6	Независими събития	62
10.7	Експеримент на Бернули	63
10.8	Проблемът за Рождения дни	64
10.9	Теорема на Бейс	65
10.10	Филтри за спам	66
11	Лекция 10: Теория на алгоритмите	68
11.1	Алгоритми	68
11.2	Сложност	68
11.3	Ръст на функция	69
11.4	Машина на Тюринг	71
12	Лекция 11: Математическа индукция	72
13	Лекция 12: Рекурсия	73
13.1	Рекурсивно дефинирани редици	73
13.2	Рекурсивни дефиниции и Индукция	73
13.3	Рекурсивни функции	73
13.3.1	Факториел	74
13.3.2	Комбинация от n елемента k -ти клас	74
13.3.3	Триъгълник на Паскал	74
13.4	Рекурсивни дефиниции	75
13.4.1	Функция най-голям общ делител (Евклид)	75
13.4.2	Математически операции	75
13.4.3	Множества	75
14	Лекция 13: Крайни автомати	76
14.1	Крайна последователност от действия	76
14.2	Азбуки и стрингове	77
14.3	Езици	77
14.4	Регулярни езици	77
14.5	Операция $*$	77
14.6	Автомати за работа със стрингове	78

Съдържание	5
------------	---

14.7 Детерминиран краен автомат (ДКА)	78
14.8 Функция на преходите	79
14.9 Недетерминиран краен автомат (НКА)	79

1 Лекция 1: Логика и логически оператори

1.1 Дефиниции

Дефиниция 1.1.1. *Логиката* е система, базирана на съждения

Дефиниция 1.1.2. *Съждението* е твърдение което може да бъде истина или лъжа (но не и двете едновременно).

Следователно резултатът от едно съждение може да бъде истина (*И*) или ако то е вярно или лъжа (*Л*), ако е грешно.

Дефиниция 1.1.3. *Съжденията*, които не съдържат в себе си други съждения, се наричат **прости**.

Дефиниция 1.1.4. Едно и няколко съждения могат да бъдат обединени в едно единствено **комбинирано съждение**, посредством логически оператори.

Дефиниция 1.1.5. *Таблица на истинност* се нарича таблица, в която се изброяват всички възможни комбинации от стойности на отделните променливи в съждението, както и съответните стойности на функцията.

Дефиниция 1.1.6. Две **съждения** са **еквиваленти**, ако имат една и съща таблица на истинност или следват едно от друго вследствие прилагани основни закони за преобразуване.

1.2 Логически оператори

1.2.1 Отрицание(NOT)

Означава се със знака \neg

Функция на една променлива с таблица на истинност:

p	$\neg p$
1	0
0	1

1.2.2 И, Конюнкция (AND)

Означава се със знака \wedge

Функция на две променливи с таблица на истинност:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

1.2.3 ИЛИ, Дизюнкция (OR)

Означава се със знака \vee

Функция на две променливи с таблица на истинност:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

1.2.4 Сума по модул 2, изключващо или (XOR)

Означава се със знака \otimes

Функция на две променливи с таблица на истинност:

p	q	$p \otimes q$
0	0	0
0	1	1
1	0	1
1	1	0

1.2.5 Импликация, следствие

Означава се със знака \rightarrow

Функция на две променливи с таблица на истинност:

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

1.2.6 Двупосочно следствие

Означава се със знака \Leftrightarrow

Функция на две променливи с таблица на истинност:

p	q	$p \Leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

1.3 Закони за еквивалентни преобразувания

- Закон за идентичност :

$$p \wedge T \equiv p \quad p \vee F \equiv p$$

- Закон за доминиране :

$$p \vee T \equiv T \quad p \wedge F \equiv F$$

- Закон за пълна идентичност :

$$p \wedge p \equiv p \quad p \vee p \equiv p$$

- Закон за двойно отрицание :

$$\neg(\neg p) \equiv p$$

- Комутативен закон :

$$p \wedge q \equiv q \wedge p \quad p \vee q \equiv q \vee p$$

- Асоциативен закон :

$$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r \quad p \vee (q \vee r) \equiv (p \vee q) \vee r$$

- Дистрибутивен закон :

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \quad p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

- Закони на Де Морган :

$$\neg(p \wedge q) \equiv (\neg p) \vee (\neg q) \quad \neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$$

- Закон за импликацията :

$$p \rightarrow q \equiv \neg p \vee q$$

- Закон за тривиалната тавтология:

$$p \vee \neg p \equiv T \quad p \wedge \neg p \equiv F$$

- Закон за тривиалното опровержение :

$$(p \Leftrightarrow q) \equiv \neg(p \otimes q) \quad \neg(p \Leftrightarrow q) \equiv (p \otimes q)$$

1.4 Предикатни функции и предикати

Дефиниция 1.4.1 (Предикатна функция). *Предикатна функция е твърдение, което съдържа една или повече променливи.*

Ако на дадена променлива е присвоена стойност, се казва, че е известна.

Предикатна функция става съждение, ако всички нейни аргументи са известни.

Дефиниция 1.4.2 (Предикат). *Нека е дадена предикатна функция $Q(x, y, z)$. Свойството Q , с което се задава връзката между променливите x , y , z се нарича предикат.*

1.5 Квантор

Нека $P(x)$ е предикатна функция.

Дефиниция 1.5.1 (Квантор за общност). *За твърдения от вида:*

За всяко x , $P(x)$ е истина/лъжа.

се записва :

$\forall x P(x)$ "за всяко x $P(x)$ "

Знака за общност е \forall .

Дефиниция 1.5.2 (Квантор за съществуване). *За твърдения от вида:*

*Съществува **такова** x , за което $P(x)$ е истина/лъжа.*

се записва :

$\exists x P(x)$ "съществува x , такова че $P(x)$ е истина/лъжа" или "Съществува поне едно x , за което $P(x)$ е истина/лъжа"

Знака за общност е \exists .

1.6 Закони на Де Морган за квантори

- $\neg(\forall P(x)) \equiv \exists x(\neg P(x))$
- $\neg(\exists P(x)) \equiv \forall x(\neg P(x))$

2 Лекция 2: Математически доказателства

2.1 Теория на доказателствата

Теорията на доказателствата се използва за определяне на верността на дадени математически аргументи и за конструирането им.

Дефиниция 2.1.1 (Дедукция). *Дедукция във философията означава извеждане на особеното и единичното от общото, както и схващане на единичния случай на базата на всеобщ закон.*

В кибернетиката означава извеждане на твърдения от други твърдения с помощта на логически заключения.

2.2 Терминология

Дефиниция 2.2.1 (Аксиома). *Аксиомата е базово допускане, което не е необходимо да се доказва. Аксиомите са твърдения, които са истина, или твърдения които се приемат за истина, но не могат да бъдат доказани.*

Дефиниция 2.2.2 (Доказателство). *Доказателството се използва, за да се докаже, че дадено твърдение е истина. То се състои от последователност от твърдения, които формират **аргумент**.*

Дефиниция 2.2.3 (Теорема). *Теорема е твърдение, чиято истинност се доказва.*

Теоремата се състои от две части: условия(хипотези) и извод(заключения). Коректното доказателство (по дедукция) се състои в това да се установи:

- *Ако условията са изпълнени, то извода е истина.*
- *Ако създението "условия \rightarrow извод" е тавтология.*

Често липват елементи на логическа връзка, която може да бъде изпълнена с допълнителни условия и аксиоми и създания, свързани помежду си посредством подходящи правила за изводи.

Дефиниция 2.2.4 (Лема). *Лемата е проста теорема, което се използва като междинен резултат за доказване на друга теорема.*

Дефиниция 2.2.5 (Слествие). *Следствието е резултат, който директно следва от съответната теорема*

Дефиниция 2.2.6 (Допускане). *Допускането е твърдение, чийто резултат е неизвестен. Веднъж доказано, то се превръща в теорема.*

2.3 Правила за извод

\therefore - знак за следователно

Всичко преди знака са хипотези.

- Събиране (Addition)

$$\begin{array}{l} p(q) \\ \therefore p \vee q \end{array}$$
- Опростяване (Simplification)

$$\begin{array}{l} p \wedge q \\ \therefore p(q) \end{array}$$
- Конюнкция (Conjunction)

$$\begin{array}{l} p \\ q \\ \therefore p \wedge q \end{array}$$
- Закон за безразличие (Modus ponens)

$$\begin{array}{l} p \\ p \rightarrow q \\ \therefore q \end{array}$$
- Modus tollens

$$\begin{array}{l} \neg q \\ p \rightarrow q \\ \therefore \neg p \end{array}$$
- Хипотетичен силогизъм (Hypothetical syllogism)

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \therefore p \rightarrow r \end{array}$$

- Дизюнктивен силогизъм

$$p \vee q$$

$$\neg p$$

$$\therefore q$$

2.4 Аргументи

Дефиниция 2.4.1 (Аргумент). *Аргументът се състои от една или няколко хипотези и заключение.*

Аргумент е валиден, ако всичките му хипотези са истина и заключението също е истина.

Ако някоя от хипотезите е лъжа, дори валиден аргумент може да води до некоректно заключение.

Доказателство: трябва да се докаже че твърдението "хипотези \rightarrow заключение" е истина, като се използват правила за извод.

Пример 2.4.1. "Ако 101 е кратно на 3, то 101^2 се дели на 9."

Въпреки че аргументът е валиден, заключението е некоректно (невярно), защото едната от хипотезите ("101 е кратно на 3") е лъжа.

Ако в аргумента 101 се замени с 102, ще се получи коректно заключение "102² е кратно на 9".

Нека p : "101 е кратно на 3" и q : " 101^2 е кратно на 9". Тогава имаме следния случай

$$p$$

$$p \rightarrow q$$

$$\therefore q$$

Обаче p е лъжа, следователно q е некоректно заключение.

2.5 Правила за извод при използване на квантори и предикати

- Универсалноследствие (Universal instantiation)

$$\forall x P(x)$$

$$\therefore P(c) \text{ ако } c \in U$$

- Универсално обобщение (Universal generalization)

$$P(c) \text{ за някое } c \in U$$

$$\therefore \forall x P(x)$$

- Частично следствие (Existential instantiation)
 $\exists x P(x)$
 $\therefore P(c)$ за някой елемент $c \in U$
- Частично обобщение (Existential generalization)
 $P(c)$ за някой елемент $c \in U$
 $\therefore \exists x P(x)$

2.6 Доказателство на теореми

- Директно доказателство
Импликацията " $p \rightarrow q$ " може да бъде доказана чрез доказване на твърдението:
"Ако p е истина, то q също е истина."
- Индиректно доказателство
Импликацията " $p \rightarrow q$ " е еквивалентна на следния контра-пример " $\neg q \rightarrow \neg p$ ". Следователно, доказателството на изходната импликация " $p \rightarrow q$ " се свежда до доказване на твърдението: "Ако q е лъжа, то и p също е лъжа."

3 Лекция 3: Теория на множества

3.1 Парадокс на Ръсел

Някои множества (класове) съдържат себе си а други не.

Ръсел нарича множествата класове.

Класът от всички класове е клас, който се съдържа (принадлежи) на себе си. Празният клас не принадлежи на себе си. Да допуснем, че може да създадем клас от всички класове, като празния например, които не съдържат себе си.

Парадоксът възниква при въпроса, дали този клас принадлежи на себе си.

Множеството от всички множества, които не съдържат себе си !?

$$M = \{A | A \notin A\}$$

- Множество – първично понятие
- Принадлежност на елемент към множеството – първично понятие
 $x \in A \rightarrow$ Елементът x принадлежи на A .
- Свойство – понятие от логиката и се приема за първично.
 P – свойство. Ако даден предмет x го притежава се записва $P(x)$

Дефиниция 3.1.1 (Множества). *Множества се дефинират като се изброяват елементите му*

$$A = \{a_1, a_2, \dots, a_n\}$$

за безкрайни множества е по - удобно като се използва свойство за принадлежност

$$R = \{x | P(x)\}$$

Аксиома 3.1.1. *Ако го има свойството P , то съществува множеството на всички обекти, които имат това свойство P .*

Аксиома 3.1.2. *Две множества са равни, ако съдържат еднакви елементи.*

Дефиниция 3.1.2 (Нормално множество). *Нормално множество е множество, което не принадлежи на себе си $A \notin A$*

3.2 Множества

- Множество е неопределена съвкупност от нула или повече различни обекти (наречени елементи).
- $a \in A$ - "а е елемент на множеството А"
- $a \notin A$ - "а не е елемент на А"
- $A = \{a_1, a_2, \dots, a_n\}$ - "А се състои от a_1, a_2, \dots, a_n "
- Подреждането на елементите не е от значение.
- Няма значение ако някой елемент се повтаря.

Пример 3.2.1. Примери за множества:

- $A = \emptyset$ - празно множество
- $A = \{z\}, z \in A, z \neq \{z\}$
- $A = \{\{b, c\}, \{c, x, d\}\}$ - множество от множества
- $A = \{\{x, y\}\}, \{x, y\} \in A, \{x, y\} \neq \{\{x, y\}\}$
- $A = \{x | P(x)\}$

3.2.1 Равенство на множества

Дефиниция 3.2.1. Множествата A и B са равни тогава и само тогава, когато съдържат едни и същи елементи.

Пример 3.2.2. $A = \{9, 2, 7, 3\}, B = \{7, 9, -3, 2\} : A = B$
 $A = \{\text{куче, котка, кон}\}, B = \{\text{котка, кон, катерица, куче}\} : A \neq B$
 $A = \{\text{куче, котка, кон}\}, B = \{\text{котка, кон, куче, куче}\} : A = B$

3.2.2 Подмножества

$A \subseteq B$ - "А е подмножество на В"

$A \subseteq B$ тогава и само тогава, когато всеки елемент на А е и елемент на В

Формален запис:

$$A \subseteq B \Leftrightarrow \forall x(x \in A \rightarrow x \in B)$$

Правила

- $A = B \Leftrightarrow (A \subseteq B) \wedge (B \subseteq A)$
- $((A \subseteq B) \wedge (B \subseteq C)) \Rightarrow (A \subseteq C)$
- $\emptyset \subseteq A$ за всяко множество A (но $\emptyset \in A$ не е вярно за всяко множество A)
- $\subseteq A$ за всяко множество A

3.2.3 Собствени подмножества

$A \subset B$ - "A е собствено подмножество на B"

$A \subseteq B$ и $A \neq B$

Формален запис:

$$A \subset B \Leftrightarrow \forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$$

или

$$A \subset B \Leftrightarrow \forall x(x \in A \rightarrow x \in B) \wedge \neg \forall x(x \in B \rightarrow x \in A)$$

3.2.4 Число на кардиналност, мощност(брой елементи) на множество

Ако множеството S съдържа n различни елемента, $n \in \mathbb{N}$, Множеството S е крайно множество с кардиналност (мощност) n .

Мощност: $|P(A)| = 2^{|A|}$

Пример 3.2.3. $A = \{Mercedes, BMW, Porsche\}, |A| = 3$

$B = \{1, \{2, 3\}, \{4, 5\}, 6\}, |B| = 4$

$C = \emptyset, |C| = 0$

$D = \{x \in \mathbb{N} | x \leq 7000\}, |D| = 7001$

$E = \{x \in \mathbb{N} | x \geq 7000\}, |E| = \infty$

3.2.5 Множество от всички подмножества на дадено множество

$P(A)$ "множеството от всички подмножества на A"(записва се като 2^A)

$P(A) = \{B | B \subseteq A\}$

3.2.6 Декартово произведение

Дефиниция 3.2.2 (Декартово произведение). Декартово произведение на две множества A и B е ново множество $A \times B$, което се дефинира като

$$A \times B = \{(a, b) | a \in A \wedge b \in B\}$$

т.е. декартовото произведение е множество от наредени двойки.

Пример 3.2.4. $A = \{x, y\}, B = \{a, b, c\}$
 $A \times B = \{(x, a), (x, b), (x, c), (y, a), (y, b), (y, c)\}$

- $A \times \emptyset = \emptyset$
- $\emptyset \times A = \emptyset$
- За непразни множества A и B : $A \neq B \Leftrightarrow A \times B \neq B \times A$
- $|A \times B| = |A| \cdot |B|$

3.3 Операции на множества

3.3.1 Обединение

$$A \cup B = \{x | x \in A \vee x \in B\}$$

Пример 3.3.1. $A = \{a, b\}, B = \{b, c, d\}$
 $A \cup B = \{a, b, c, d\}$

3.3.2 Сечение

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

Несвързани множества: Няма общи елементи, $A \cap B = \emptyset$

Пример 3.3.2. $A = \{a, b\}, B = \{b, c, d\}$
 $A \cap B = \{b\}$

3.3.3 Разлика

$$A \setminus B = \{x | x \in A \wedge x \notin B\}$$

Пример 3.3.3. $A = \{a, b\}, B = \{b, c, d\}$
 $A \setminus B = \{a\}$

3.3.4 Допълнение

$$A^c \equiv \overline{A} = U - A$$

U - универсално множество.

Пример 3.3.4. $U = N$

$$B = \{250, 251, 252, \dots\}$$

$$\overline{B} = \{0, 1, 2, \dots, 248, 249\}$$

3.4 Идентичност на законите за логическо преобразуване и законите за операции с множества

- Закон за идентичност :

$$A \cup \emptyset = A$$

$$A \cap U = A$$

- Закон за доминиране :

$$A \cup U = U$$

$$A \cap \emptyset = \emptyset$$

- Закон за пълна идентичност :

$$A \cup A = A$$

$$A \cap A = A$$

- Закон за двойно отрицание :

$$\overline{\overline{A}} \equiv (A^c)^c = A$$

- Комутативен закон :

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

- Асоциативен закон :

$$A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C$$

- Дистрибутивен закон :

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

3.4 Идентичност на законите за логическо преобразуване и законите за операции с множества 20

- Закони на Де Морган :

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

- Закон за поглъщането :

$$A \cup (A \cap B) = A$$

$$A \cap (A \cup B) = A$$

- Закон за допълнението:

$$A \cup \overline{A} = U$$

$$A \cap \overline{A} = \emptyset$$

4 Лекция 4: Релации (Отношения)

4.1 Релации

Дефиниция 4.1.1 (Бинарна релация). Нека A и B са две множества. Бинарна релация от A в B е подмножество на $A \times B$.

С други думи, за дадена бинарна релация R е в сила $R \subseteq A \times B$. Означението aRb означава $(a, b) \in R$, а означението $a \not R b$ означава, че $(a, b) \notin R$.

Когато $(a, b) \in R$ се казва, че a е в отношение с b чрез R .

Пример 4.1.1. Нека P е множество от хора, C е множество от коли, а D е релацията, която описва кой човек каква кола (коли) кара.

$P = \{\text{Иван, Лили, Петър, Валя}\},$

$C = \{\text{Опел, BMW, Рено}\}$

$D = \{(\text{Иван, Опел}), (\text{Лили, Опел}), (\text{Лили, Рено}), (\text{Петър, BMW})\}$

4.2 Функциите като релации

Графиката на f е множество от наредени двойки (a, b) , такова че $b = f(a)$. Тъй като графиката на f е подмножество на $A \times B$, то тя е релация от A в B .

Още повече, за всеки елемент на A , съществува точно една наредена двойка в графиката на функцията, чийто първи елемент е a .

Обратно, ако R е релация от A в B , такава че всеки елемент от A е първи елемент на само една наредена двойка от R , тогава функцията може да бъде дефинирана посредством R като нейна графика.

Това се прави като на всеки елемент $a \in A$ се присвоява само един елемент $b \in B$, така че $(a, b) \in R$.

4.3 Релации, дефинирани върху множество

Дефиниция 4.3.1. Релация, дефинирана върху множество A , е релация от A в A .

С други думи, дадена релация, дефинирана върху множество A , е подмножество на $A \times A$.

Пример 4.3.1. Нека $A = \{1, 2, 3, 4\}$. Кои наредени двойки са от релацията $R = \{(a, b) | a < b\}$

Решение: $R = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$

Колко различни релации могат да бъдат дефинирани върху едно множество A с n елемента?

Броят на елементите на $A \times A = n^2$, тогава броя на подмножествата (релации върху A) е 2^{n^2} - броят на елементите на множеството от всички подмножества на $A \times A$.

Отговор: Броят на различните релации, дефинирани върху A , е 2^{n^2}

4.4 Свойства на релациите

Дефиниция 4.4.1 (Рефлексивна). Релацията R , дефинирана върху A , се нарича **рефлексивна**, ако $(a, a) \in R$ за всеки елемент $a \in A$

Дефиниция 4.4.2 (Антирефлексивна). Релацията R , дефинирана върху A , се нарича **антирефлексивна**, ако $(a, a) \notin R$ за всеки елемент $a \in A$

Дефиниция 4.4.3 (Симетрична). Релацията R , дефинирана върху A , се нарича **симетрична**, ако $(b, a) \in R$ когато $(a, b) \in R$ за $a, b \in A$.

Дефиниция 4.4.4 (Антисиметрична). Релацията R , дефинирана върху A , се нарича **антисиметрична**, ако когато $(a, b) \in R$ и $(b, a) \in R$ то $a = b, a, b \in A$.

Дефиниция 4.4.5 (Асиметрична). Релацията R , дефинирана върху A , се нарича **асиметрична**, ако когато $(a, b) \in R$ и $(b, a) \notin R$ за $a, b \in A$.

Дефиниция 4.4.6 (Транзитивна). Релацията R , дефинирана върху A , се нарича **транзитивна**, ако когато $(a, b) \in R$ и $(b, c) \in R$, тогава $(a, c) \in R$ за $a, b, c \in A$.

Пример 4.4.1. Релации, дефинирани върху $\{1, 2, 3, 4\}$

$R = \{(1, 1), (2, 2), (2, 3), (3, 3), (4, 4)\}$ - рефлексивна

$R = \{(1, 1), (1, 2), (2, 1), (3, 3), (4, 4)\}$ - симетрична

$R = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$ - антисиметрична

$R = \{(1, 3), (3, 2), (2, 1)\}$ - асиметрична

$R = \{(1, 1), (1, 2), (2, 2), (2, 1), (3, 3)\}$ - транзитивна

4.5 Операции с релациите

Релациите са множества, и затова, върху тях може да бъдат прилагани стандартни **операции с множества**.

Ако R_1, R_2 са релации от множеството A в множеството B , тогава може да разгледаме операциите $R_1 \cup R_2, R_1 \cap R_2, R_1 - R_2$

Във всеки от случаите, резултатът от операциите е друга релация от A в B .

4.5.1 Комбиниране на релациите

Дефиниция 4.5.1. Нека R е релация от множеството A в множеството B и S е релация от множеството B в множеството C .

Композицията от R и S е релация състояща се от наредени двойки $(a, c); a \in A, c \in C$ и за които съществува елемент $b \in B$ такива че $(a, b) \in R$ и $(b, c) \in S$

Композицията се означава $S \circ R$.

Пример 4.5.1. Нека D и S са релации от $A = \{1, 2, 3, 4\}$

$$D = \{(a, b) | b = 5 - a\}$$

$$S = \{(a, b) | a < b\}$$

$$D = \{(1, 4), (2, 3), (3, 2), (4, 1)\}$$

$$S = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$$

$$S \circ R = \{(2, 4), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)\}$$

$$S \circ R = \{(a, b) | b > 5 - a\} \Leftrightarrow S \circ R = \{(a, b) | a + b > 5\}$$

Дефиниция 4.5.2. Нека R е релация дефинирана върху множеството A . Степените на R при получаване на композиция, $R^n, n = 1, 2, 3, \dots$, се дефинират по индукция като

$$R^1 = R$$

$$R^{n+1} = R^n \circ R$$

$$R^n = R \circ R \circ \dots \circ R \text{ (} n\text{-пъти } R\text{)}$$

Теорема 4.5.1. Релацията R върху множеството A е транзитивна тогава и само тогава когато $R^n \subseteq R$ за всяко цяло положително число n .

4.6 n-арни релации

Дефиниция 4.6.1. Нека A_1, A_2, \dots, A_n са множества. n -арна релация върху тези множества е подмножеството $A_1 \times A_2 \times \dots \times A_n$. Множествата A_1, A_2, \dots, A_n се наричат области (домейни) на релацията, а n - ред на релацията.

Пример 4.6.1. Нека $R = \{(a, b, c) | a = 2b \wedge b = 2c; a, b, c \in \mathbb{N}\}$

Редът на R - 3, тъй като елементите са тройки.

Области на R - множеството на естествените числа \mathbb{N}

$(2, 4, 8)$ - не е от R

$(4, 2, 1)$ - е от R

4.7 Бази данни и релации

Дефиниция 4.7.1. Базата данни се състои от n -торки наречени **записи**, които са изградени от **атрибути (полета)**. Тези полета (атрибути) са **позициите** в n -торките.

Релационният модел на данните представя базата данни като n -арна релация, което по същество представлява множество от записи, като всеки от **атрибутите (полетата)** има своя **област (домейн)**.

Пример 4.7.1. Разглеждаме база данни от студенти, чиито записи представляват наредени 4-творки с полета (атрибути): Име, Фак.номер, Факултет, и Успех:

$R = \{(\text{Иван}, 231455, \text{ФА}, 4.88), (\text{Георги}, 888323, \text{ФЕТТ}, 4.45), (\text{Димитър}, 232147, \text{ФА}, 5.29), (\text{Петър}, 453876, \text{ФКСУ}, 5.45), (\text{Николай}, 458543, \text{ФКСУ}, 4.45), (\text{Стефан}, 886576, \text{ФЕТТ}, 5.18)\}$

Прост код: ако n -торките могат да бъдат точно определени по стойностите на това поле. Това означава, че няма два (или повече) записа с една и съща стойност на полето, което е прост код.

Дефиниция 4.7.2 (Проекция). Проекцията P_{i_1, i_2, \dots, i_m} изобразява n -торката (a_1, a_2, \dots, a_n) в m -торка $(a_{i_1}, a_{i_2}, \dots, a_{i_m})$ където $m \leq n$

Пример 4.7.2. Какъв резултат се получава, ако се приложи проекцията $P_{2,4}$ на записа от базата данни:

(Георги, 888323, ФЕТТ, 4.45)? Отговор: Наредена двойка (888323, 4.45)

Дефиниция 4.7.3 (Съединение). Нека R е релация от ред m а S е релация от ред n . Съединение $J_p(R, S)$, където $p \leq m$ и $p \leq n$, е релация от ред $m + n - p$, която съдържа всички $(m + n - p)$ -торки

$$(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$$

такива че m -торката $(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p)$ принадлежи на R , а n -торката $(c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$ принадлежи на S .

Пример 4.7.3. Да се определи съединението $J_1(Y, R)$, където релацията Y съдържа атрибутите Год. на раждане и Име.

$Y = \{(1982, \text{Иван}), (1980, \text{Георги}), (1984, \text{Димитър}), (1979, \text{Петър}), (1983, \text{Николай}), (1985, \text{Стефан})\}$

$R = \{(\text{Иван}, 231455, \text{ФА}, 4.88), (\text{Георги}, 888323, \text{ФЕТТ}, 4.45), (\text{Димитър}, 232147, \text{ФА}, 5.29), (\text{Петър}, 453876, \text{ФКСУ}, 5.45), (\text{Николай}, 458543, \text{ФКСУ}, 4.45), (\text{Стефан}, 886576, \text{ФЕТТ}, 5.18)\}$

Решение:

$J_1 = \{(1982, \text{Иван}, 231455, \text{ФА}, 4.88), (1980, \text{Георги}, 888323, \text{ФЕТТ}, 4.45), (1984, \text{Димитър}, 232147, \text{ФА}, 5.29), (1979, \text{Петър}, 453876, \text{ФКСУ}, 5.45), (1983, \text{Николай}, 458543, \text{ФКСУ}, 4.45), (1985, \text{Стефан}, 886576, \text{ФЕТТ}, 5.18)\}$

Тъй като Y има два атрибута, R има четири атрибута, а общият атрибут Име е един, релацията $J_1(Y, R)$ има $2 + 4 - 1 = 5$ атрибута.

4.8 Представяне релации

Два нови начина за представяне на релации: булеви матрици и ориентирани графи.

4.8.1 Булеви матрици

Ако R е релация от $A = \{a_1, a_2, \dots, a_m\}$ в $B = \{b_1, b_2, \dots, b_n\}$, то R може да бъде представена посредством булева матрица (матрица с елементи единици и нули)

$M_R = [m_{ij}]$ с елементи

$m_{ij} = 1, (a_i, b_j) \in R,$

$m_{ij} = 0, (a_i, b_j) \notin R.$ За формиране на матрицата трябва първо да се изредят елементите на A и B в определен (но произволен) ред.

Пример 4.8.1. Да се представи релацията $R = \{(2, 1), (3, 1), (3, 2)\}$ като булева матрица?

Решение:

$$M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Матрици, които представят релации върху множество трябва да бъдат квадратни матрици.

Матрици, които представят рефлексивни релации всички елементи от главния диагонал на такива матрици M_{ref} трябва да са единици.

Матрици, които представят симетрични релации трябва да бъдат симетрични матрици, $M_R = (M_R)^T$.

При определяне на матрицата, която представя обединение на две релации се прилага логическата операция “или” върху съответните елементи на матриците, представящи релациите.

При определяне на матрицата, която представя сечение на две релации се прилага логическата операция “и” върху съответните елементи на матриците, представящи релациите.

Пример 4.8.2. Нека релациите R и S се представят с матриците:

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Определете матриците, които представят $R \cup S$ и $R \cap S$?

Решение:

$$M_{R \cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad M_{R \cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Дефиниция 4.8.1. Нека $A = [a_{ij}]$ е $m \times k$ булева матрица, а $B = [b_{ij}]$ е $k \times n$ булева матрица.

Булевото произведение на A и B , се означава с $A \circ B$, и е $m \times n$ матрица (i, j) -тият елемент $[c_{ij}]$, на която е

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \dots \vee (a_{ik} \wedge b_{kj})$$

$c_{ij} = 1$ тогава и само тогава, когато поне един от компонентите $(a_{in} \vee b_{nj})$ е 1 за някое n ; $c_{ij} = 0$ в останалите случаи.

4.8.2 Ориентирани графи

Дефиниция 4.8.2 (Ориентиран граф). *Ориентираният граф, се състои от множество V от върхове (или възли) и множество E от наредени двойки от V наречено ребра (или дъги). Върха се нарича начален връх на ребро (a, b) , а връх b се нарича краен връх на реброто. Стрелките се използват за означаване на посоките.*

Очевидно всяка релация R върху A може да се представи с ориентиран граф, като елементите на A са негови върхове, а всички наредени двойки $(a, b) \in R$ са негови ребра.

Обратно, всеки ориентиран граф с върхове V и ребра E може да се представи с релация върху V , съдържаща всички наредени двойки от E .

Това взаимно еднозначно съответствие между релации и ориентирани графи означава, че всяко твърдение за релациите е приложимо за ориентираните графи и обратно.

4.9 Релации на еквивалентност

Дефиниция 4.9.1 (Релация на еквивалентност). *Дадена релация, дефинирана върху множеството A , се нарича релация на еквивалентност ако е рефлексивна, симетрична и транзитивна.*

Два елемента, свързани посредством дадена релация на еквивалентност R , се наричат еквивалентни.

Тъй като R е симетрична, a е еквивалентен на b , когато b е еквивалентен на a .

Тъй като R е рефлексивна, всеки елемент е еквивалентен на себе си.

Тъй като R е транзитивна, ако a и b са еквивалентни и b и c са еквивалентни, то a и c са еквивалентни.

Очевидно, тези три свойства са необходими за обосновано дефиниране на еквивалентността.

Пример 4.9.1. Нека R е релация върху множество от думи на български език, такава че aRb тогава и само тогава, когато $l(a) = l(b)$, където $l(x)$ е дължината на думата x .

Релация на еквивалентност ли е R ? Решение:

R е рефлексивна, защото $l(a) = l(a)$ и следователно aRa за всяка дума a .

R е симетрична, защото ако $l(a) = l(b)$ то $l(b) = l(a)$, така че ако aRb то

bRa .

R е транзитивна, защото ако $l(a) = l(b)$ и $l(b) = l(c)$, то $l(a) = l(c)$, така че от aRb и bRc , следва че aRc .

Следователно R е релация на еквивалентност.

4.10 Класове на еквивалентност

Дефиниция 4.10.1 (Клас на еквивалентност). Нека R е релация на еквивалентност дефинирана върху множеството A . Множеството от всички елементи свързани с даден елемент " a " от A посредством R се нарича клас на еквивалентност на " a ".

Класът на еквивалентност на " a " по отношение на R се означава с $[a]_R$. Ако $b \in [a]_R$, b - представител.

Пример 4.10.1 (котка). е множеството от всички думи на български език с пет букви. Например, "петел" е представител на този клас на еквивалентност.

Теорема 4.10.1. Нека R е релация на еквивалентност, дефинирана върху A . Долните твърдения са еквивалентни:

- aRb
- $[a] = [b]$
- $[a] \cap [b] \neq \emptyset$

Дефиниция 4.10.2 (Разбиване). Разбиванена множеството S е съвкупност от несвързани, непразни подмножества от S , чието обединение е S .

С други думи, съвкупността от подмножества $A_i, i \in I$, формират разделяне на S тогава и само тогава, когато

- $A_i \neq \emptyset, i \in I$
- $A_i \cap A_j = \emptyset, i \neq j$
- $\bigcup_{i \in I} A_i = S$

Пример 4.10.2. Нека S е $\{u, m, b, r, o, c, k, s\}$.

Разбиване на S ли са долните съвокупности от множества?

$\{\{m, o, c, k\}, \{r, u, b, s\}\}$ - да.

$\{\{c, o, m, b\}, \{u, s\}, \{r\}\}$ - не(k липсва).

$\{\{b, r, o, c, k\}, \{m, u, s, t\}\}$ - не(t не е от S).

$\{\{u, m, b, r, o, c, k, s\}\}$ - да.

$\{\{b, o, o, k\}, \{r, u, m\}, \{c, s\}\}$ - да ($\{b, o, o, k\} = \{b, o, k\}$).

$\{\{u, m, b\}, \{r, o, c, k, s\}, \emptyset\}$ не(\emptyset не е позволено).

Теорема 4.10.2. Нека R е релация на еквивалентност, дефинирана върху S . Тогава класовете на еквивалентност, свързани с R , формират разбиване на S .

Обратно, за дадено разбиване $\{A_i | i \in I\}$ на множеството S , съществува релация на еквивалентност R , за която множествата $A_i, i \in I$, са класове на еквивалентност.

Пример 4.10.3. Нека Иван, Силвия и Георги живеят в София, Мая и Петър живеят в Пловдив, а Кирил живее в Бургас.

Нека R е релация на еквивалентност $\{(a, b) \mid a \text{ и } b \text{ живеят в един и същ град}\}$ върху множеството $P = \{\text{Иван, Силвия, Георги, Мая, Петър, Кирил}\}$.

Тогава

$R = \{(\text{Иван, Иван}), (\text{Иван, Силвия}), (\text{Иван, Георги}), (\text{Силвия, Иван}), (\text{Силвия, Силвия}), (\text{Силвия, Георги}), (\text{Георги, Иван}), (\text{Георги, Силвия}), (\text{Георги, Георги}), (\text{Мая, Мая}), (\text{Мая, Петър}), (\text{Петър, Мая}), (\text{Петър, Петър}), (\text{Кирил, Кирил})\}$.

Класовете на еквивалентност, свързани с R , са: $\{\{\text{Иван, Силвия, Георги}\}, \{\text{Мая, Петър}\}, \{\text{Кирил}\}\}$. Това също е и разбиване на R .

Пример 4.10.4. Нека R е релацията $\{(a, b) \mid a \equiv b \pmod{3}\}$, дефинирана върху множеството на целите числа.

Релация на еквивалентност ли е R ?

Да, защото R е рефлексивна, симетрична и транзитивна.

Кои са класовете на еквивалентност свързани с R ?

$\{\{..., -6, -3, 0, 3, 6, ...\}\},$

$\{\{..., -5, -2, 1, 4, 7, ...\}\},$

$\{\{..., -4, -1, 2, 5, 8, ...\}\}$

5 Лекция 5: Функции

Функция f от множеството A в множеството B е такова съответствие, че на всеки елемент от множеството A е съпоставен (присвоен) един (единствен) елемент от множеството B .

Присвояването се записва като

$$f(a) = b$$

където b е единствения елемент от B присвоен посредством f на елемента a от A . Ако f е функция от множество A в множество B , то това се означава по следния начин

$$f : A \rightarrow B$$

Множеството A е дефиниционна област на f , а множеството B е област на стойностите на f .

b е образ на a , а a е първообраз на b .

Диапазон на f е множеството от всички образи на всички елементи на A .

Пример 5.0.1. Да разгледаме функцията $f : P \rightarrow C$ където

$P = \{\text{Лили, Митко, Катя, Петър}\}$

$C = \{\text{Бургас, София, Пловдив, Варна}\}$

$f(\text{Лили}) = \text{Пловдив}$

$f(\text{Митко}) = \text{София}$

$f(\text{Катя}) = \text{Бургас}$

$f(\text{Петър}) = \text{Варна}$

За случая, диапазонът на f е C .

Начини за представяне на функциите :

- таблици

x	$f(x)$
Лили	Пловдив
Митко	София
Катя	Бургас
Петър	Варна

- означаване на съответствията със стрелки
- формули $f : \mathbb{R} \rightarrow \mathbb{R}, \quad f(x) = 2x$
- графики: множеството от наредени двойки $\{(a, b) | a \in A, f(a) = b\}$.

Нека f_1 и f_2 са функции от A в \mathbb{R} . Сумата и произведението на f_1 и f_2 също са функции от A в \mathbb{R} дефинирани като:

$$(f_1 + f_2)(x) = f_1(x) + f_2(x)$$

$$(f_1 f_2)(x) = f_1(x) f_2(x)$$

Пример 5.0.2.

$$f_1(x) = 3x \quad f_2(x) = x + 5$$

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = 3x + x + 5 = 4x + 5$$

$$(f_1 f_2)(x) = f_1(x) f_2(x) = 3x(x + 5) = 3x^2 + 15x$$

Ако разгледаме подмножеството $S \subseteq A$, то множеството от всички образи на елементите $s \in S$ се нарича образ на S .

Образът на S се означава с $f(S)$: $f(S) = \{f(s) | s \in S\}$

5.1 Функциите като релации

Тъй като графиката на f е подмножество на $A \times B$, то тя е релация от A в B .

Още повече, за всеки елемент на A , съществува точно една наредена двойка в графиката на функцията, чийто първи елемент е a .

Обратно, ако R е релация от A в B , такава че всеки елемент от A е първи елемент на само една наредена двойка от R , тогава функцията може да бъде дефинирана посредством R като нейна графика.

Това се прави като на всеки елемент $a \in A$ се присвоява само един елемент $b \in B$, така че $(a, b) \in R$

5.2 Свойства на функции

5.2.1 Инекция

Функцията $f : A \rightarrow B$ е "изображение в" или инекция (влагане), тогава и само тогава, когато

$$\forall x, y \in A (f(x) = f(y) \rightarrow x = y)$$

5.2.2 Строго нарастваща/намаляваща

Функцията $f : A \rightarrow B$, където $A, B \subseteq \mathbb{R}$ се нарича строго нарастваща, ако

$$\forall x, y \in A (x < y \rightarrow f(x) < f(y))$$

и строго намаляваща, ако

$$\forall x, y \in A (x < y \rightarrow f(x) > f(y))$$

5.2.3 Сюрекция

Функцията $f : A \rightarrow B$ е "изображение върху" или сюрекция (налагане), тогава и само тогава, когато за всеки елемент $b \in B$ съществува елемент $a \in A$, така че $f(a) = b$.

5.2.4 Биекция

Функцията $f : A \rightarrow B$ е взаимно еднозначно съответствие, или биекция, тогава и само тогава, когато е едновременно инекция и сюрекция.

5.3 Обратна функция

Обратна функция на биекцията $f : A \rightarrow B$ е функцията

$$f^{-1} : B \rightarrow A, f^{-1}(b) = a \Leftrightarrow f(a) = b.$$

5.4 Крайни и безкрайни множества

Дефиниция 5.4.1. Множеството A е крайно, ако $A \neq \emptyset$ или $\exists n \in \mathbb{N}$ и биекция $f : A \rightarrow I_n$. Мощността $|A| = 0$, ако $A = \emptyset$.

Дефиниция 5.4.2. Множеството A е изброимо безкрайно, ако съществува биекция $f : A \rightarrow \mathbb{N}$. Множеството A е изброимо, ако е крайно или изброимо безкрайно.

Индексацията на елементите на едно множество A с елементите на друго множество е знак за съществуване на биекция между тях.

Дефиниция 5.4.3. Едно множество е безкрайно ако е равномощно на свое собствено подмножество.

Пример 5.4.1. Функцията $f : N \rightarrow N_2, f(n) = 2n$ изобразява множеството на естествените числа N в собственото му подмножество на четните естествени числа N_2 . Следователно N е безкрайно множество. Множеството $A = \{1, 2, 3\}$ е крайно, тъй като не съществува биекция между него и негово собствено подмножество.

5.5 Принцип на Дирихле

Ако X и Y са крайни множества, чиито мощности са в отношение $|X| > |Y|$, то за всяка функция $f : X \rightarrow Y$, съществуват $a_1 \neq a_2 \in X$, за които е в сила $f(a_1) = f(a_2)$.

Принципът на Дирихле неформално се нарича принципа на чекмеджетата. Ако множеството X е множество от предмети а Y е множество от чекмеджета, така че предметите са повече от чекмеджетата и всички предмети от X се поставят в чекмеджетата от Y , то поне в едно чекмедже ще има поне два предмета.

5.6 Композиция

Композицията на две функции $g : A \rightarrow B, f : B \rightarrow C$, се означава с $f \circ g$, и се дефинира като:

$$(f \circ g)(a) = f(g(a))$$

Това означава :

- първо, функцията g се прилага на елемента $a \in A$, като го изобразява в елемент от B
- второ, функцията f се прилага на този елемент от B , изобразявайки го в елемент от C .
- Следователно, съставната функция $(f \circ g)$ е изображение от A в C .

Композиция на функция с обратната и функция:

$$(f^{-1} \circ f)(x) = f^{-1}(f(x)) = x$$

Пример 5.6.1.

$$\begin{aligned}
f : \mathbb{R} &\rightarrow \mathbb{R}, & g : \mathbb{R} &\rightarrow \mathbb{R} \\
f(x) &= 7x - 4, & g(x) &= 3x \\
(f \circ g)(5) &= f(g(5)) = f(15) = 105 - 4 = 101 \\
(f \circ g)(x) &= f(g(x)) = f(3x) = 21x - 4
\end{aligned}$$

5.7 Функции най-голямо цяло (Floor) и най-малко цяло (Ceiling)

Функциите floor и ceiling изобразяват реалните числа в целите числа $\mathbb{R} \rightarrow \mathbb{Z}$.

Функцията най-голямо цяло (floor) присвоява на дадено $r \in \mathbb{R}$ най-голямото цяло число $z \in \mathbb{Z}$, такова че $z \leq r$.

Означава се с $\lfloor r \rfloor$.

По същество това е закръгляване надолу.

Пример 5.7.1.

$$\lfloor 2.3 \rfloor = 2, \quad \lfloor 2 \rfloor = 2, \quad \lfloor 0.5 \rfloor = 0, \quad \lfloor -3.5 \rfloor = -4$$

Функцията най-малко цяло (ceiling) присвоява на дадено $r \in \mathbb{R}$ най-малкото цяло число $z \in \mathbb{Z}$, такова че $z \geq r$.

Означава се с $\lceil r \rceil$.

По същество това е закръгляване нагоре.

Пример 5.7.2.

$$\lceil 2.3 \rceil = 3, \quad \lceil 2 \rceil = 2, \quad \lceil 0.5 \rceil = 1, \quad \lceil -3.5 \rceil = -3$$

6 Лекция 6: Булева алгебра

При булевата алгебра операциите и правилата за работа с тях са свързани с множеството $\{0, 1\}$.

6.1 Булеви операции

Дефиниция 6.1.1. *Отрицанието се означава с $-$. Дефинира се по следния начин*

$$-0 = 1 \quad -1 = 0$$

Дефиниция 6.1.2. *Булевата сума, се означава с $+$ и правилата за прилагането и са следните:*

$$1 + 1 = 1 \quad 1 + 0 = 1 \quad 0 + 1 = 1 \quad 0 + 0 = 0$$

Дефиниция 6.1.3. *Булевото умножение, се означава с \cdot , и правилата за прилагането му са следните:*

$$1 \cdot 1 = 1 \quad 1 \cdot 0 = 0 \quad 0 \cdot 1 = 0 \quad 0 \cdot 0 = 0$$

6.2 Булеви (двоични) функции и изрази

Дефиниция 6.2.1. *Нека $B = \{0, 1\}$. Променливата x се нарича булева променлива ако приема стойности само от B .*

Дадена функция от B^n , т.е. $\{(x_1, x_2, \dots, x_n) | x_i \in B, 1 \leq i \leq n\}$, в B се нарича булева (двоична) функция от ред n .

Булевите функции могат да бъдат представени посредством: формули, които са изрази от булеви променливи и булеви операции и таблици със стойности на функцията за всяка комбинация от стойности на променливите.

Дефиниция 6.2.2. *Булев израз на променливите x_1, x_2, \dots, x_n може да се дефинира рекурсивно както следва:*

$$0, 1, x_1, x_2, \dots, x_n \text{ са булеви изрази}$$

$$E_1, E_2 \rightarrow \text{булеви изрази, то } (-E_1), (E_1 E_2), (E_1 + E_2) \text{ са булеви изрази}$$

Всеки булев израз представя булева функция.

Стойностите на тази функция могат да бъдат намерени чрез заместване на булевите променливи в израза с 0 и 1.

Пример 6.2.1. Намерете формулата (булевият израз) за булевата функция $F(x, y)$, която се дефинира с долната таблица:

x	y	F(x,y)
0	0	0
0	1	1
1	0	0
1	1	0

Възможно решение: $F(x, y) = (-x) \cdot y$

Най-простият метод за получаване на формула (булев израз) за функция зададена с таблица от стойности на функцията за всяка комбинация от стойности на променливите се базира на булева сума от компоненти. Тези компоненти съответстват на комбинациите от стойности на променливите, за които функцията има стойност 1

Те са булево произведение от вида $y_1 y_2 \dots y_n, y_i = x_i (x_i = 1)$ или $y_i = -x_i (x_i = 0)$

Дефиниция 6.2.3. Булевите функции F и G на n променливи са равни тогава и само тогава, когато $F(b_1, b_2, \dots, b_n) = G(b_1, b_2, \dots, b_n)$ за b_1, b_2, \dots, b_n от B .

Два различни булеви изрази (формули), които представят една и съща функция се наричат еквивалентни.

Например, булевите изрази $xy, xy + 0, xy \cdot 1$ са еквивалентни.

Дефиниция 6.2.4. Булево допълнение или отрицание на булевата функция F е булевата функция $-F$, където

$$-F(b_1, b_2, \dots, b_n) = -(F(b_1, b_2, \dots, b_n))$$

Дефиниция 6.2.5. Нека F и G са булеви функции от ред n . Булева сума $F + G$ и булево произведение $F \cdot G$ се дефинират като:

$$(F + G)(b_1, b_2, \dots, b_n) = F(b_1, b_2, \dots, b_n) + G(b_1, b_2, \dots, b_n)$$

$$(F \cdot G)(b_1, b_2, \dots, b_n) = F(b_1, b_2, \dots, b_n) \cdot G(b_1, b_2, \dots, b_n)$$

Броят на различните булеви функции от ред n е 2^{2^n} .

6.3 Дуалност

Дуален на даден булев израз се получава като се заменят булевите суми с булеви умножения, булевите умножения с булеви суми, единиците се заменят с нули и нулите с единици.

Тази дуална функция, се означава с F^d , и не зависи от конкретния булев израз използван за представяне на F .

Пример 6.3.1. Дуалният израз на $x(y + z) = x + yz$
 Дуалният израз на $-x \cdot 1 + (-y + z) = (-x + 0) \cdot ((-y)z)$

Равенството между функции представени с булеви изрази остава валидно когато се вземат дуалните функции на двете страни на равенството.

Това се нарича принцип на дуалността, и може да се използва за извеждане на нови равенства.

Пример 6.3.2. Закон за поглъщането

$$x(x + y) = x$$

Като вземем дуалните изрази на двете страни на равенството, се получава

$$x + xy = x$$

което също е равенство (и също се нарича закон за поглъщането).

6.4 Дефиниция на булевата алгебра

Дефиниция 6.4.1. Булева алгебра е множество B с елементи 0 и 1 , две бинарни операции $(+ \text{ и } \cdot)$ и една унарна операция $(-)$, така че за всяко x, y, z от B са в сила следните свойства :

- *Закони за идентичност*

$$x + 0 = x \quad x \cdot 1 = x$$

- *Закони за доминация*

$$x + (-x) = 1 \quad x \cdot (-x) = 0$$

- *Асоциативни закони*

$$(x + y) + z = x + (y + z) \quad (x \cdot y) \cdot z = x \cdot (y \cdot z)$$

- *Комутативни закони*

$$x + y = y + z \quad x \cdot y = y \cdot x$$

- *Дистрибутивни закони*

$$x + (y \cdot z) = (x + y) \cdot (x + z) \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

6.5 Логика, множества и булева алгебра

Логика	Множества	Булева алгебра
Лъжа	\emptyset	0
Истина	U	1
$A \wedge B$	$A \cap B$	$A \cdot B$
$A \vee B$	$A \cup B$	$A + B$
$\neg A$	A^c	$\overline{A}(-A)$

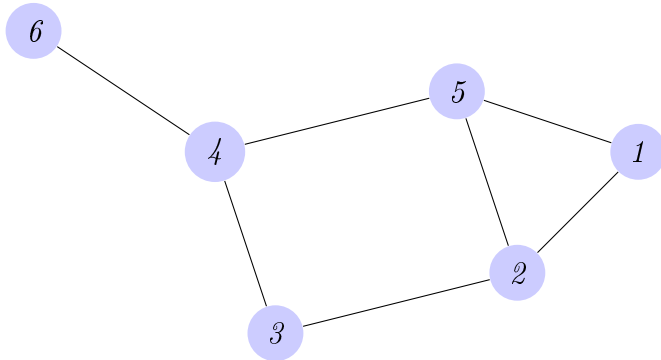
7 Лекция 7: Графи

7.1 Дефиниции и терминология

Дефиниция 7.1.1. *Простия граф $G = (V, E)$ е математическа структура, която се състои от:*

- Непразно множество от върхове V .
- Множество E от ненаредени двойки от различни елементи от V , наречени ребра
- Всяко ребро $e \in E$ е множество $= \{u, v\}$ където $u, v \in V$ (м/у 2 върха има само едно ребро)
- Даден ненасочен (неориентиран) граф може да съдържа примки (тогава не е прост).
- Дадено ребро е примка ако $e = \{u, u\}$ за някое $u \in V$.

Прост граф:



Дефиниция 7.1.2. *Мултиграф $G = (V, E)$ е такъв граф, който може да съдържа примки и двойки върхове свързани с няколко ребра.*

Дефиниция 7.1.3. *Даден граф е планарен, ако може да бъде начертан в една равнина без да има пресичане на ребрата му.*

Дефиниция 7.1.4. *Насоченият (ориентираният) граф $G = (V, E)$ е математическа структура, която се състои от:*

- Непразно множество от върхове V .

- Множество E от наредени двойки от различни елементи от V , наречени ребра
- Всяко ребро $e \in E$ е наредена двойка $= (u, v)$ където $u, v \in V$
- Дадено ребро е примка ако $e = (u, u)$ за някое $u \in V$.

Дефиниция 7.1.5. Два върха u, v в ненасочен граф G се наричат съседни в G ако $\{u, v\}$ е ребро в G .

С цел определяне на отношението на принадлежност на ребрата на графа към върховете му и обратно се въвежда понятието инцидентност.

Дефиниция 7.1.6. Ако $e = \{u, v\}$, реброто e се нарича инцидентно с върховете u и v .

Реброто e свързва u и v .

Върховете u и v се наричат крайни точки на реброто $\{u, v\}$.

Дефиниция 7.1.7. Степен (валентност) на даден връх от неориентиран граф е броят на ребрата свързани с него.

Степента на връх v се означава с $\deg(v)$.

Връх със степен 0 се нарича изолиран, тъй като не е съседен на никой друг връх на графа.

Връх с примка е от степен поне 2 и по дефиниция не е изолиран, дори да не е съседен на друг връх.

Връх със степен 1 се нарича висящ. Такъв връх е съседен на точно един друг връх от графа.

Теорема 7.1.1. Нека $G = (V, E)$ е ненасочен граф с n ребра. Тогава

$$2n = \sum_{v \in V} \deg(v)$$

Пример 7.1.1. Колко са ребрата в граф с 10 върха, всеки от степен 6.

Решение: Сумата от степените на върховете на графа е $6 \cdot 10 = 60$. От горната теорема, следва че $2n = 60$, т.е. $n = 30$ ребра.

Теорема 7.1.2. Всеки ненасочен граф има четен брой върхове от нечетна степен.

Дефиниция 7.1.8. Когато (u, v) е ребро в ориентиран граф G , върховете u и v са съседни.

Върхът u се нарича *аначален върх* на реброто (u, v) а върхът v се нарича *краен върх* на (u, v) .

Началният и крайният върх на дадена примка съвпадат.

Дефиниция 7.1.9. В ориентиран граф, полустепенята на върх v , се означава с $\deg^-(v)$, и е равна на броят на ребрата, чиито краен върх е v .

Полустепенната върх v , означена с $\deg^+(v)$, е броят на ребрата, чиито начален върх е v .

Теорема 7.1.3. Нека $G = (V, E)$ е ориентиран граф. Тогава:

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

7.2 Специални графи

Дефиниция 7.2.1. Пълен граф е прост граф, който съдържа точно едно ребро между всяка двойка различни върхове.

Пълен граф с n върха се означава с K_n .

Дефиниция 7.2.2. Граф-контур $C_n, n \geq 3$ се състои от n върха v_1, v_2, \dots, v_n и ребрата $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.

Дефиниция 7.2.3. Граф-колело W_n се получава ако към даден граф-контур $C_n, n \geq 3$, се добави допълнителен върх, който се свързва с всички върхове на контура C_n с нови ребра.

Дефиниция 7.2.4. Граф n -куб, се означава с Q_n , и е граф, чиито върхове представляват $2n$ стринга от битове с дължина n . Два върха са съседни (свързани с ребро) тогава и само тогава, когато стринговете от битове, които които представляват върховете се различават само по един бит.

Дефиниция 7.2.5. Даден прост граф се нарича *двуделен* ако множеството на върховете му V може да се разбие на две непресичащи се множества V_1, V_2 , така че всяко ребро в графа свързва върхове от V_1 с върхове от V_2 (по този начин няма ребро в графа, което свързва два върха само от V_1 или само от V_2).

Пример 7.2.1. Графът, чрез който всеки човек от даден град се представя с връх а всеки брак с ребро.

Този граф е двуделен, защото всяко ребро свързва върхове от две подмножества от мъжете и жените (има се предвид нормални бракове).

Дефиниция 7.2.6. *Пълен двуделен граф $K_{m,n}$ е граф, чиито върхове са разделени на две подмножества съответно от m и n елементи. Два върха са свързани с ребро тогава и само тогава ако са от различни подмножества.*

Дефиниция 7.2.7. *Подграф на даден граф $G = (V, E)$ е граф $H = (W, F)$ където $W \subseteq V, F \subseteq E$.*

Дефиниция 7.2.8. *Обединение на два ненасочени графа $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ е ненасочен граф с множество от върховете $V_1 \cup V_2$ и множество от ребрата $E_1 \cup E_2$. Обединението на G_1 и G_2 се означава с $G_1 \cup G_2$.*

7.3 Представяне на графите

Дефиниция 7.3.1. *Нека $G = (V, E)$ е неориентиран граф и $|V| = n$. Нека върховете на G са подредени в произволен ред v_1, v_2, \dots, v_n .*

Матрица на съседство A на G при избраната подредба на върховете е $n \times n$ матрица от единици и нули с 1 в позиция (i, j) , когато v_i и v_j са съседни, и 0 в противен случай. В случаите на мултиграфи, различните от нула елементи показват колко ребра свързват съответните върхове.

Пример 7.3.1. Матрицата на съседство A_G за графа G при следния ред на върховете a, b, c, d е

$$A_G = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Дефиниция 7.3.2. *Нека $G = (V, E)$ е неориентиран граф и $|V| = n$. Нека върховете и ребрата на G са подредени в произволен ред v_1, v_2, \dots, v_n и e_1, e_2, \dots, e_m .*

Матрица на инцидентност на G при избраната подредба на върховете и ребрата е $n \times m$ матрица от единици и нули с 1 в позиция (i, j) , когато ребро e_j е инцидентно с връх v_i и 0 в противен случай.

Пример 7.3.2. Матрицата на инцидентност M за графа G при следната подредба на върховете a, b, c, d и ребрата 1, 2, 3, 4, 5, 6 е

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

7.4 Изоморфизъм на графи

Дефиниция 7.4.1. Два ненасочени графа $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ са изоморфни ако съществува биекция (взаимно еднозначно съответствие) $f : V_1 \rightarrow V_2$, такава че a и b са съседни в G_1 , тогава и само тогава когато $f(a)$ и $f(b)$ са съседни в G_2 , за всяко a и b от V_1 . Такава функция f се нарича изоморфизъм.

Визуално, G_1 и G_2 са изоморфни ако могат да бъдат организирани по такъв начин, че техните изображения да са еднакви.

За два прости графа с по n върха има $n!$ възможни изоморфизми, които трябва да бъдат проверени, за да се покаже, че графите са изоморфни. За да може два графа да са изоморфни трябва да се провери инвариантността, на някои параметри на графите, което е задължително свойство. Двата графа трябва да имат:

- един и същ брой върхове,
- един и същ брой ребра,
- едни и същи степени на върховете.

Ако горните параметри не са инвариантни, то графите не са изоморфни, но ако два графа имат инвариантни параметри, те не са задължително изоморфни.

7.5 Пътища и контури в графи

Дефиниция 7.5.1. Път с дължина n от u до v , където n е положително цяло число, в **ненасочен граф** е последователност от ребра e_1, e_2, \dots, e_n на графа, така че $e_1 = \{x_0, x_1\}, e_2 = \{x_1, x_2\}, \dots, e_n = \{x_{n-1}, x_n\}$, където $x_0 = u$ и $x_n = v$.

Даден път е контур (затворен път) ако започва и свършва с един и същ връх, т.е. $u = v$.

Дефиниция 7.5.2. Път с дължина n от u до v , където n е положително цяло число, в **насочен граф** е последователност от ребра e_1, e_2, \dots, e_n на графа, така че $e_1 = \{x_0, x_1\}, e_2 = \{x_1, x_2\}, \dots, e_n = \{x_{n-1}, x_n\}$, където $x_0 = u$ и $x_n = v$.

Даден път е контур (затворен път) ако започва и свършва с един и същ връх, т.е. $u = v$.

Дефиниция 7.5.3. Казва се, че даден път или контур преминава през върховете x_1, x_2, \dots, x_{n-1} .

Дефиниция 7.5.4. Даден път или контур е прост ако не съдържа едно ребро два пъти.

Дефиниция 7.5.5 (Ойлеров контур). Даден контур е Ойлеров, ако преминава през всяко ребро само по един път.

Дефиниция 7.5.6 (Хамилтонов контур). Даден контур е Хамилтонов, ако преминава през всеки връх само по един път.

7.6 Свързаност

Дефиниция 7.6.1. Даден ненасочен граф е свързан ако съществува път между всяка двойка ребра.

Дефиниция 7.6.2. Даден несвързан граф е обединение на два или повече свързани подграфа, като всяка двойка от тези подграфи няма общи върхове. Тези подграфи се наричат свързани компоненти на графа.

Дефиниция 7.6.3. Даден прост граф е дърво ако не съдържа контури и кратни ребра (няколко ребра, които свързват два върха). Броят на ребрата на дървото е $n-1$, където n е броят на върховете на графа.

Дефиниция 7.6.4. Даден насочен граф е силно свързан ако съществува път от a до b и от b до a , за всяка двойка върхове a и b от графа.

Дефиниция 7.6.5. Даден насочен граф е слабо свързанако има път между всеки два върха на съответния ненасочен граф.

7.7 Достижимост

Дефиниция 7.7.1. Нека $G = (V, E)$ е неориентиран граф и $|V| = n$. Нека върховете на G са подредени в произволен ред v_1, v_2, \dots, v_n . Матрица на достижимост за p стъпки H_p на G при избраната подредба на върховете е $n \times n$ матрицата

$$H_p = I \cup A_G \cup A_G \cdot A_G \cup \dots \cup A_G \cdot A_G \dots A_G (p \text{ пъти умножение})$$

Пример 7.7.1. Матрицата на достижимост за 2 и 3 стъпки за графа G при следния ред на върховете a, b, c, d

$$A_G = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad H_2 = ? \quad H_3 = ?$$

$$A_G^2 = A_G \cdot A_G = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H_2 = I \cup A_G \cup A_G^2 =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cup \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cup \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$A_G^3 = A_G^2 \cdot A_G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H_3 = I \cup A_G \cup A_G^2 \cup A_G^3 =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cup \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cup \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cup \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

7.8 Графи с допълнителни характеристики на ребрата

На ребрата от даден граф може да бъдат присвоени тегла, свързани с допълнителни характеристики на обекта, който се моделира с дадения граф. Например при моделиране на дадена железопътна мрежа (в аспект на разстояния), разстоянията между отделните градове могат да служат като тегла на съответните ребра от графа.

В общ случай тези характеристики могат да бъдат цени, пропускателни способности на линии и др.

7.9 Най-къс път

Графи с тегла на ребрата могат да служат за моделиране на компютърни мрежи, като теглата могат да бъдат свързани с време за реакция, цена и др.

Най-интересния въпрос за този тип графи е:

Какъв е най-късият път между два върха в графа, в смисъл такъв път, за който сумата от теглата на ребрата от пътя е минимална?

С тази задача от теорията на графите се моделира проблема за най-късото разстояние по железопътната мрежа между два града или на най-бърза връзка в компютърна мрежа.

7.10 Задача на търговския пътник

Дадени са n върха. Колко различни граф-контура C_n могат да бъдат формирани, свързвайки върховете с ребра?

Отговор: Да отбележим, че всеки връх от граф-контура може да бъде считан за начален. Ако фиксираме даден връх, то тогава имаме $(n - 1)$ възможности за избор на втори връх от граф-контура, $(n - 2)$ възможности за избор на трети връх, и т.н. По този начин имаме $(n - 1)!$ възможности за формиране на граф-контури. Разгледаният начин за получаване на граф-контурите е свързан с посока на обхождане от началния до последния възел. Поради това, горният брой възможности включват същите контури но при обратна посока на обхождане. Следователно, ре-

алният брой граф-контури $C_n = \frac{(n - 1)!}{2}$.

8 Лекция 8: Дървета

8.1 Дървета

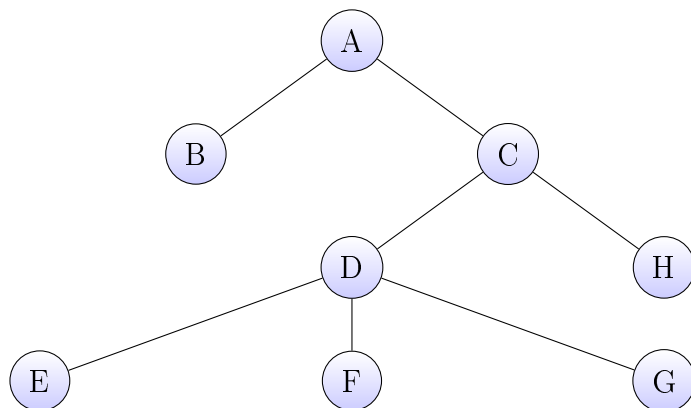
Дефиниция 8.1.1. *Дървото е прост свързан граф, в който няма контури (затворени пътища).*

Дървото не съдържа примки и между всяка двойка върхове може да има само едно ребро (няма кратни ребра).

Дефиниция 8.1.2. *Гората е граф, в който всеки свързан компонент е дърво.*

Теорема 8.1.1. *Даден граф е дърво тогава и само тогава, когато между всеки два върха има само един път.*

Дефиниция 8.1.3. *Ако даден връх бъде определен за корен на дървото, то е възможно да бъде въведена посока на всяко от ребрата.*



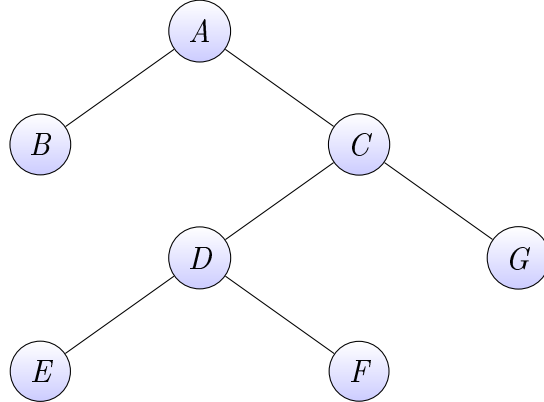
Видове върхове

- Корен - А ("начало")
- Вътрешен връх - С (има родител и деца)
- Лист - В (няма деца)
- Родител - С е родител на D (връх на по горно ниво)
- Прародител - А е прародител на D (родителя на родителя)
- Деца - Е, F, G (имат родител, но нямат самите те деца)

Дефиниция 8.1.4. *Едно дърво с корен е t -арно дърво ако всеки вътрешен връх има не повече от t върха деца.*

Дефиниция 8.1.5. *Дървото е пълно t -арноако всеки вътрешен връх има точно t върха деца.*

t -арно дърво с $t=2$ се нарича бинарно дърво.



8.2 Дървета с корен и наредба на върховете

Дефиниция 8.2.1. *Дърво с корен и наредба на върховете е дърво с корен, за което върховете деца на всеки вътрешен връх са подредени. Дърветата с наредба се чертаят, така че върховете деца на всеки вътрешен връх се подреждат отляво наясно.*

Теорема 8.2.1. *Дадено дърво с n върха има $n - 1$ ребра.*

Теорема 8.2.2. *Дадено пълно t -арно дърво с i вътрешни върха има общо $n = t \cdot i + 1$ върхове.*

8.3 Свойства на дърветата

Дадено пълно t -арно дърво с

1. n върха има $i = \frac{n-1}{t}$ вътрешни върхове и $l = \frac{(t-1)(n+1)}{t}$ листа
2. i вътрешни върха има $n = t \cdot i + 1$ върха и $l = (t-1)i + 1$ листа
3. l листа има $n = \frac{(ml-1)}{t-1}$ върха и $i = \frac{l-1}{t-1}$ вътрешни върха

Дефиниция 8.3.1. *Нивото на даден връх v в дърво с корене дължината на единствения път от корена до този връх v .*

Дефиниция 8.3.2. *Височината на дадено дърво с корен е равна на максималното ниво на върховете.*

Дефиниция 8.3.3. *m -арно дърво с корен и височина h се нарича балансирано ако всичките му листа са на нива h или $h - 1$.*

Дефиниция 8.3.4. *В m -арно дърво с височина h има най-много m^h листа.*

Теорема 8.3.1. *Ако m -арно дърво с височина h има l листа, то $h \geq \lceil \log_m l \rceil$*

8.4 Покриващо дърво

Покриващо дърво на даден граф е подграф, който съдържа всички върхове и е дърво (свързан без контури).

Един граф може да има много покриващи дървета, например за графа от фигурата.

8.5 Минимално покриващо дърво (МПД)

Минимално покриващо дърво (МПД) е подграф на неориентиран граф $G = (V, E)$ с тегла на ребрата, така че:

1. той е дърво (т.е. свързан без цикли)
2. състои се от всички върхове от множеството V и съдържа $|V| - 1$ ребра
3. общата сума от теглата на ребрата от покриващото дърво е минимална, по отношение на останалите покриващи дървета
4. може да има няколко покриващи дървета

Лема 8.5.1. *Нека X е подмножество от върхове на графа G , и нека реброто e е реброто с най-малко тегло, което свързва X с $G - X$ (върховете, които не принадлежат на X).*

Тогавата e е част от МПД

8.6 Метод на Крускал

Нека елементите на минималното покриващо дърво (МПД) на графа $G = \{V, E\}$ формират множеството A .

1. Първоначално $A = \emptyset$.
2. Подреждат се елементите на множеството $E' = \{e_1, e_2, \dots, e_m\}$ от ребра на графа по ненамаляващ ред на съответстващите им тегла в ново множество E' .
3. Завсеки елемент (ребро) (u, v) от подреденото множество E' , движейки се по нарастващ ред на номера на съответния елемент, се проверява дали той може да принадлежи на покриващото дърво, т.е. да не се допуска формиране на затворен контур.
 - При "да" разглежданото ребро (u, v) се добавя към покриващото дърво (елемент от множеството A)
 - При "не" се преминава към анализ на следващия по нарастващ ред на номера елемент от подреденото множество E' .

Критерият за спиране е когато всички върхове от анализираният неориентиран граф се включат в множеството A , т.е. принадлежат на търсеното МПД.

8.7 Метод на Прим

Нека елементите на минималното покриващо дърво (МПД) на графа $G = \{V, E\}$ формират множеството A .

1. Първоначално $A = \emptyset$.
2. Избира се произволен връх v_i от графа
3. Избира се реброто с най-малък тегловен коефициент, свързано с върха v_i , и се проверява дали то може да принадлежи на МПД, т.е. да не се допуска формиране на затворен контур от ребра принадлежащи на МПД.
 - При "да" разглежданото ребро се добавя към множеството A .

- При "не" се преминава към анализ на реброто със следващия по нарастващ ред тегловен коефициент, свързано с върха v_i .
4. От множеството от ребра свързани с върхове на ребрата от A се избира немаркирано (непринадлежащо на A) ребро с най-малък тегловен коефициент. Се проверява дали то може да принадлежи на МПД, т.е. да не се допуска формиране на затворен контур от ребра принадлежащи на покриващото дърво.
- При "да" разглежданото ребро се добавя към множеството A .
 - При "не" се преминава към анализ на реброто със следващия по нарастващ ред тегловен коефициент, свързано с някой от върховете на ребрата от A .
5. Стъпка 4 се повтаря, докато всички върхове от анализираният граф се включат в множеството A , т.е. принадлежат на търсеното МПД.

8.8 Особености на методите на Крускал и Прим

1. Метод на Крускал

- Като резултат от метода се получава множество от ребра A , което формира "гора".
- Дадено ребро се прибавя като елемент на множеството A винаги, когато на него съответства най-малкият по стойност все още неанализиран тегловен коефициент на ребро от графа, свързващо два несвързани върха.

2. Метод на Прим

- Като резултат от метода се получава множество от ребра A , което формира просто дърво.
- Дадено ребро се прибавя като елемент на множеството A винаги, когато то е свързано с анализирания възел и на него съответства най-малкият по стойност тегловен коефициент на всички ребра, излизащи от разглеждания възел на дървото.

9 Лекция 9: Комбинаторика

9.1 Принцип на Дирихле

Принцип на зайците и чекмеджетата:

Ако $k+1$ или повече обекти се намират в k кутии.

Тогава има поне една кутия, съдържаща два или повече от обектите.

Пример 9.1.1. Няколко примера:

- Измежду която и да е група от 367 човека, трябва да има поне двама с една и съща рождена дата.
- В която и да е група от 27 английски думи, трябва да има поне две, които да започват с една и съща буква.
- Колко хора трябва да бъдат избрани от 15 семейни двойки, за да сме сигурни, че поне двама от избраните са женени един за друг?

9.2 Принцип на добавянето

Да предположим, че имаме k задачи.

Нека първата задача може да бъде решена по n_1 начина, втората - по n_2 начина и т.н. k -тата задача - по n_k начина.

Тези задачи не трябва да бъдат решавани едновременно. т.е. трябва да бъде решена или първата, или втората, и т.н. или последната задача.

Тогава има $n_1 + n_2 + \dots + n_k$ различни начина за решение на проблема.

Пример 9.2.1. Студент може да избира компютърен проект от един от три списъка. Трите списъка съдържат съответно 23, 15 и 19 възможни проекта. От колко възможни проекта може да избира студентът?

Всички проекти са различни.

Следователно, общо $23 + 15 + 19 = 57$ проекта.

Разглеждане от гледна точка на теорията на множествата:

Ако A_1, A_2, \dots, A_k са несвързани множества (т.е. всички от елементите са различни), тогава броя на елементите в обединението на тези множества е сумата от броя елементи в тях.

$$|A_1 \cup A_2 \cup \dots \cup A_k| = |A_1| + |A_2| + \dots + |A_k|$$

Пример 9.2.2. Колко цели числа между 1 и 100 (включително 100) са четни или завършват на 5?

A - множеството на четните числа между 1 и 100. $|A| = 50$

B - множеството от цели числа, завършващи на 5. $|B| = 10$

Елементите от A и B са различни, защото числото, завършващо на 5 не може да бъде четно.

Така че, според принципа на добавяне

$$|A \cup B| = |A| + |B| = 50 + 10 = 60$$

9.3 Принцип на умножението

Да предположим, че процедура е съставена от последователност от k стъпки.

Ако първата стъпка може да бъде изпълнена по n_1 начина, втората стъпка по n_2 начина и изобщо i-тата стъпка по n_i начина.

Тогава има $n_1 \cdot n_2 \cdot \dots \cdot n_i$ начина, за да изпълним процедурата.

Пример 9.3.1. Колко са различни стрингове от нули и единици с дължина седем?

Работим с битове със стойност или единица, или нула.

Всеки от седемте бита може да бъде избран по два начина.

Така че общо има $2^7 = 128$ различни стринга.

Разглеждане от гледна точка на теорията на множествата:

Ако A_1, A_2, \dots, A_k са крайни множества, тогава броят елементи на Декартовото произведение от тези множества е произведението от броя на елементите във всяко множество.

$$|A_1 \times A_2 \times \dots \times A_k| = |A_1| \cdot |A_2| \cdot \dots \cdot |A_k|$$

9.4 Принцип на включването и изключването

Принципът на добавянето не може да бъде използван, ако множествата са свързани.

Ако множества A и B са свързани, тогава трябва да използваме принципа на включването и изключването:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

9.5 Дървовидни структури за изброяване

Дървовидна структура е графично средство, което може да помогне за разрешаването на проблеми, свързани с преброяване.

Дървото се състои от

- Корен
- Брой клони, излизащи от корена
- Възможни допълнителни клони, излизащи от края на другите кло-ни
- Листа: краищата на клоните, които нямат други клони, започващи от тях.

Използваме клони, за да представим всички възможни избори.

9.6 Пермутации и комбинации

Модели за избор на обекти от урна (Urn models):

Дадено е множество от n обекта в една урна.

Вземаме (избираме) r обекта от урната последователно.

След като сме избрали обект може или да го върнем обратно в урната - (избор с връщане) или да не го връщаме в урната - (избор без връщане).

Колко различни възможни поредици от r обекта са възможни?

Има ли значение подредбата на обектите или не?

При избор с връщане имаме n^r възможни поредици.

При избор без връщане имаме два подпроблема: колко различни наредби (Пермутация) и колко начина на избор (Комбинация) имаме.

9.6.1 Пермутации

Представлява наредба или последователност от n различни обекта.

Ако $r \leq n$, тогава наредбата, при която се използват r от n различни обекта се нарича r -пермутация.

Броят различни r -пермутации от множество от n различни елемента се означава с $P(n, r)$.

$$P(n, r) = n(n-1)(n-2)\dots(n-r+1) = \frac{n!}{(n-r)!}$$

9.6.2 Комбинации

Това е избор без подредба измежду n различни обекта.

Ако $r \leq n$, тогава неопределен избор на r измежду n различни обекта се нарича r -комбинация.

Броят различни r -комбинации от множества от n различни елемента се означава с $C(n, r)$.

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Тъждество на Паскал:

$$C(n+1, k) = C(n, k-1) + C(n, k)$$

10 Лекция 9А: Теория на вероятностите

10.1 Опит, събитие, вероятност

- Събитие: Протичането на дадено явление.
- Случайно събитие: Такова събитие, което при даден комплекс от условия може да се сбъдне, а може и да не се сбъдне.
- Теорията на вероятностите: Занимава се с изследване на закономерностите при случайните събития.
- Опит (експеримент): Комплекс от действия, извършени при строго определени условия.
- Елементарни изходи: В резултат от опита се случва точно един от възможните изходи. Тези възможни изходи от даден опит се наричат (елементарни събития –ще ги означаваме с ω).
- Пространство на елементарните изходи: Множеството от всички елементарни изходи, което обикновено се означава с Ω .

Пример 10.1.1. Хвърляне на зар:

Пространство на елементарните изходи/събития: $S = \{1, 2, 3, 4, 5, 6\}$.

Събитие: Всяко подмножество на пространството на елементарните събития: $E_1 = \{4\}$, $E_2 = \{1, 6\}$

- Събитията се изразяват чрез елементарни събития.
- Елементарните събития не могат да се изразят чрез други.
- Сигурно (достоверно) събитие (означава се с U): Ако при многократно повтаряне на даден комплекс от условия се реализира само това събитие.
- Невъзможно събитие (означава се с V): Ако при многократно повтаряне на даден комплекс от условия такова събитие не се реализира (сбъдва).
- Противоположни събития: Реализират се при един и същи комплекс от условия и ако винаги, когато едното събитие се реализира, то другото не се реализира.

- Несъвместими събития: Събития, които не могат да се реализират (едновременно) при един и същ комплекс от условия, се наричат несъвместими. Множествата на елементарните им събития са несвързани.
- Съвместими събития: Събития, които не са несъвместими.

Пример 10.1.2. Хвърляне на зар:

Пространство на елементарните изходи/събития: $S = \{1, 2, 3, 4, 5, 6\}$.

Събитие: Всяко подмножество на пространството на елементарните събития: $E_1 = \{4\}$, $E_2 = \{1, 6\}$.

Вероятност за събитие E :

$$P(E) = \frac{\text{броя на елементите на } E}{\text{броя на елементите на } S} = \frac{|E|}{|S|}, \quad 0 \leq p(E) \leq 1$$

Пример 10.1.3. Хвърляне на зар:

Пространство на елементарните изходи/събития: $S = \{1, 2, 3, 4, 5, 6\}$.

- Каква е вероятността, че при едно хвърляне на зара се пада 1?
Събитие: $E = \{1\}$
Вероятност: $p(E) = \frac{|E|}{|S|} = \frac{1}{6}$
- Каква е вероятността, че при едно хвърляне на зара се пада 2 или 5? Събитие: $E = \{2, 5\}$
Вероятност:

$$P(E) = \frac{|E|}{|S|} = \frac{2}{6} = \frac{1}{3}$$

Пример 10.1.4. Хвърляне на два еднакви зара:

Пространство на елементарните изходи/събития:

$$S = \{(1, 1), (1, 2), (1, 3), \dots, (6, 6)\}$$

- Каква е вероятността при хвърляне на два зара да се получи комбинацията (1,1)?
Събитие: $E = \{(1, 1)\}$
Вероятност: $p(E) = \frac{|E|}{|S|} = \frac{1}{36}$

- Каква е вероятността при хвърляне на два зара да се паднат едни и същи числа? Събитие: $E = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}$
Вероятност:

$$P(E) = \frac{|E|}{|S|} = \frac{6}{36} = \frac{1}{6}$$

Пример 10.1.5. Игра с неопределени числа:

Организаторът на играта избира множеството от 6 печеливши числа от общо 40 числа

всички числа: 1, 2, 3, ..., 40

печеливши числа: 4, 7, 16, 25, 33, 39

Играчът избира набор от 6 числа (редът им е без значение)

Каква е вероятността играчът да е избрал същите числа като организатора (печалба)?

Събитие: $E = \{\{4, 7, 16, 25, 33, 39\}\}$, $|E| = 1$

Пространство на елементарните изходи: $S = \{\text{всички подмножества с 6 числа от общо 40}\}$, $|S| = \binom{40}{6} = 3838380$

Вероятност:

$$P(E) = \frac{1}{\binom{40}{6}} = \frac{1}{3838380}$$

Пример 10.1.6. Игра на карти:

Едно тесте съдържа 52 карти: 13 вида карти (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A) и всеки вид има 4 бои (Пика, спатия, купа, каро).

Играчът получава ръка от 4 карти. Каква е вероятността всички карти на играча да са от един и същ вид? Събитие: $E = \{\text{всеки комплект от 4 карти от един и същи вид}\}$. $|E| = 13$

Пространство на елементарните изходи: $S = \{\text{всички подмножества с 4 карти от общо 52}\}$, $|S| = \binom{52}{4} = \frac{52!}{4! \cdot 48!} = 270725$

Вероятност:

$$P(E) = \frac{13}{270725}$$

Пример 10.1.7. Игра с подредени числа:

Организаторът на играта избира 5 топки в определен ред, от кофа с топки маркирани с числата от 1 до 50

всички числа: 1, 2, 3, ..., 50

печеливши числа: 37, 4, 16, 33, 9

Играчът избира съвкупност от 5 топки маркирани с числа (редът на

числата е от значение)

числа на играча: 40,16,13,25,33

Каква е вероятността играчът да е избрал същите числа като организатора (успех)?

Събитие: $E = \{\{40, 16, 13, 25, 33\}\}, |E| = 1$

Пространство на елементарните изходи: Ако топките не се връщат $|S| = P(50, 5) = \frac{50!}{(50-5)!} = 245251200$

Вероятност:

$$P(E) = \frac{1}{245251200}$$

Пространство на елементарните изходи: Ако топките се връщат $|S| = 50^5 = 312500000$

Вероятност:

$$P(E) = \frac{1}{312500000}$$

10.2 Вероятност на противоположно събитие

Вероятност на противоположно събитие:

$$P(\overline{E}) = 1 - P(E)$$

Пример 10.2.1. Каква е вероятността един двоичен стринг от 8 бита да съдържа поне една нула?

$E = \{01111111, 10111111, \dots, 00111111, \dots, 00000000\}$

$\overline{E} = \{11111111\}$

$$P(E) = 1 - P(\overline{E}) = 1 - \frac{|\overline{E}|}{|S|} = 1 - \frac{1}{2^8}$$

10.3 Вероятност на обединение от събития

Вероятност на обединение от събития: $E_1, E_2 \subseteq$

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$$

Пример 10.3.1. Каква е вероятността един двоичен стринг от 8 бита да започва с 0 или да завършва на 11?

$E_1 = \{00000000, 00000001, \dots, 01111111, \dots\}, |E_1| = 2^7$

$E_2 = \{00000011, 00000111, \dots, 11111111\}, |E_2| = 2^6$

$$|E_1 \cap E_2| = 2^5$$

Вероятност:

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$$

$$\frac{|E_1|}{|S|} + \frac{|E_2|}{|S|} - \frac{|E_1 \cap E_2|}{|S|}$$

$$\frac{2^7}{2^8} + \frac{2^6}{2^8} - \frac{2^5}{2^8} = \frac{1}{2} + \frac{1}{4} - \frac{1}{8} = \frac{5}{8}$$

10.4 Теория на вероятностите

Множество на елементарните изходи:

$$S = \{x_1, x_2, \dots, x_n\}$$

Функция на разпределение на вероятността p

$$0 \leq P(x_i) \leq 1$$

$$\sum_{x=1}^n P(x_1) = 1$$

Пример 10.4.1. Монета, при която вероятността за падане на лице или герб е различна.

Ези (лице) с вероятност $2/3$.

Тура(герб) с вероятност $1/3$.

Пространство на изходите $S = \{E, T\}$

$$P(E) = \frac{2}{3} \quad P(T) = \frac{1}{3} \quad P(E) + P(T) = \frac{2}{3} + \frac{1}{3} = 1$$

10.4.1 Равномерно разпределениена вероятностите

Равномерно разпределениена вероятностите:

$$P(x_i) = \frac{1}{n}$$

Пространство на елементарните изходи:

$$S = \{x_1, x_2, \dots, x_n\}$$

Пример 10.4.2. Монета с еднаква вероятност за падане на лице или герб
 "Лице" или "герб" с вероятности $1/2$

$$S = \{E, T\} \quad P(E) = \frac{1}{2} \quad P(T) = \frac{1}{2}$$

10.4.2 Вероятност на събитие

$$E = \{x_1, x_2, \dots, x_k\} \subseteq S$$

$$P(E) = \sum_{i=1}^k P(x_i)$$

За равномерно разпределение на вероятността:

$$P(E) = \frac{|E|}{|S|}$$

Пример 10.4.3. Зар с нееднаква вероятностна падане на различните страни.

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$P(1) = P(2) = P(3) = P(4) = P(5) = \frac{1}{7}, \quad P(6) = \frac{2}{7}$$

Каква е вероятността изходът от жребия да бъде 2 или 6?

$$E = \{2, 6\}$$

$$P(E) = P(2) + P(6) = \frac{1}{7} + \frac{2}{7} = \frac{3}{7}$$

10.4.3 Обединение на несъвместими събития

$$P\left(\bigcup_i E_i\right) = \sum_i P(E_i)$$

10.5 Условна вероятност

Означаване на събитие при предварително зададено условие

$$E_F = E|F$$

събитие E , при дадено условие F

$$P(E_F) = P(E|F) = \frac{P(E \cap F)}{P(F)}$$

Пример 10.5.1. Три хвърляния на монета с еднаква вероятност за падане на лице или герб.

Условие: Първата монета е паднала на тура.

Въпрос: Каква е вероятността да има нечетен брой падания на тура, при условие че първата монета е паднала на тура?

$$S = \{TTT, TTE, TET, TEE, ETT, ETE, EET, EEE\}$$

Условие на ограничено пространство на елементарните изходи. Първата монета е паднала на герб

$$F = \{TTT, TTE, TET, TEE\}$$

Събитие, получено без предварително условие, а само да има нечетен брой падане на герб

$$E = \{TTT, TEE, ETE, EET\}$$

$$E_F = E \cap F = \{TTT, EEE\}$$

$$P(E_F) = \frac{|E \cap F|}{|F|} = \frac{|E \cap F|/|S|}{|F|/|S|} = \frac{P(E \cap F)}{P(F)} = \frac{2/8}{4/8} = 0.5$$

10.6 Независими събития

Събития, при които настъпването на всяко от тях, не влияе на вероятността за настъпване на другото.

За тях е в сила правилото, известно като теорема за умножение на вероятностите.

Събитията E_1, E_2 са независими ако

$$P(E_1 \cap E_2) = P(E_1)P(E_2)$$

Ако $P(E_2) > 0$

$$P(E_1|E_2) = P(E_1)$$

Пример 10.6.1. 4-битови стрингове със случайноразпределение на битовете.

E_1 : стринг започващ с 1

E_2 : стринг съдържащ четен брой единици

$$E_1 = \{1111, 1110, 1101, 1100, 1011, 1010, 1001, 1000\}, \quad |E_1| = 8$$

$$E_2 = \{1111, 1100, 1010, 1001, 0110, 0101, 0011, 0000\} \quad |E_2| = 8$$

$$E_1 \cap E_2 = \{1001, 1010, 1100, 1111\}, \quad |E_1 \cap E_2| = 4$$

$$P(E_1) = P(E_2) = \frac{8}{16} = \frac{1}{2}$$

$$P(E_1 \cap E_2) = \frac{4}{16} = \frac{1}{4} = \frac{1}{2 \cdot 2} = P(E_1)P(E_2)$$

10.7 Експеримент на Бернули

Експеримент на Бернули:

Експеримент с два възможни резултата:

Успех или неуспех (провал)

Вероятност за успех: p

Вероятност за неуспех: $q = 1 - p$

Пример 10.7.1. Монета с различна вероятност западане на ези или тура.

Успех = ези. $p = P(E) = \frac{1}{3}$.

Неуспех = тура. $q = P(T) = \frac{2}{3}$.

Независими експерименти на Бернули:

Резултатите от последователните опити на Бернули не зависят един от друг.

Пример 10.7.2. Последователно хвърляне на монети.

Хвърляне на монета с нееднаква вероятност за падане ези или тура, 5 пъти:

Каква е вероятността монетата да падне 3 пъти ези?

Вероятност за падане ези $p = \frac{2}{3}$, успешен.

Вероятност за падане тура $q = \frac{1}{3}$, неуспешен.

Общ брой начини за подреждане в последователност от 5 хвърляния на

монета с 3 пъти падане на лице: $\binom{5}{3}$.

Вероятност, че която и да е конкретна последователност съдържа 3 лица и 2 герба на определени позиции: p^3q^2 .

Вероятност да се падне 3 пъти лице:

$$p^3q^2 + p^3q^2 + \dots + p^3q^2 = \binom{5}{3}p^3q^2$$

Вероятност да се падне точно 3 пъти ези

$$\binom{5}{3}p^3q^2 = \frac{5!}{3!2!} \left(\frac{2}{3}\right)^3 \left(\frac{1}{3}\right)^2 \approx 0.0086$$

Теорема 10.7.1. *Вероятност да се получат k успешни изхода от n независими експерименти на Бернули:*

$$C(n, k)p^kq^{n-k} = \binom{n}{k}p^kq^{n-k}$$

Познато като биномиално разпределение на вероятността

$$b(k; n, p) = C(n, k)p^kq^{n-k} = \binom{n}{k}p^kq^{n-k}$$

Пример 10.7.3. Случайни равномерни двоични стрингове.

Вероятността за нулев бит е 0.9

Вероятността за единичен бит е 0.1

Каква е вероятността да имаме 8 нулеви бита от общо 10 бита?

$$p = 0.9 \quad q = 0.1 \quad k = 8 \quad n = 10$$

$$b(k : n, p) = \binom{n}{k}p^kq^{n-k} = \binom{10}{8}(0.9)^8(0.1)^2 = 0.19137102445$$

10.8 Проблемът за Рождените дни

Съвпадение на рождени дни: двама души да имат рожден ден на една и съща дата.

Колко души трябва да има в една стая, така че вероятността за съвпадение на рождени дни да е поне $1/2$?

Предположение: равна вероятност да бъдат родени в който и да е ден от

годината.

В една година има 366 дни.

Ако броят на хората е 367 или повече, тогава съвпадението на рождени дни е гарантирано съгласно принципа за "гълъбите и кутиите".

Да предположим, че имаме $n < 366$ души.

p_n - вероятност, че всички n хора имат различни рождени дни.

$1 - p_n$ - вероятност, че съществува съвпадение на рождени дни между n на брой души.

$$S = \{1, 2, \dots, 366\} \times \{1, 2, \dots, 366\} \times \dots \times \{1, 2, \dots, 366\}, |S| = 366^n$$

$$E = \{(1, 2, \dots, 366), (366, 1, \dots, 365), \dots, (366, 365, \dots, 1)\}, |E| = P(366, n) = \frac{366!}{(366 - n)!}$$

$$p_n = \frac{|E|}{|S|} = \frac{366 \cdot 365 \cdot \dots \cdot (366 - n + 1)}{366^n}$$

$$n = 22 \quad 1 - p_n \approx 0.475$$

$$n = 23 \quad 1 - p_n \approx 0.506$$

За $n \geq 23$ души има вероятност поне $1/2$ за съвпадение на рождените дни.

10.9 Теорема на Бейс

$$P(E) \neq 0 \quad P(F) \neq 0$$

$$P(F|E) = \frac{P(E|F)P(F)}{P(E|F)P(F) + P(E|\bar{F})P(\bar{F})}$$

Пример 10.9.1. Случаен избор на кутия, последващ случаен избор на топка от кутията.

Кутия 1: 2 зелени и 7 червени.

Кутия 2: 4 зелени и 3 червени.

Ако е избрана червена топка, каква е вероятността тя да е била избрана от първата кутия?

E: Избор на червена топка

:Избор на зелена топка

F:Избрана е първата кутия

\bar{F} :Избрана е втората кутия

$P(F|E) = ?$

Теорема на Бейс:

$$P(F|E) = \frac{P(E|F)P(F)}{P(E|F)P(F) + P(E|\bar{F})P(\bar{F})}$$

$$P(F) = \frac{1}{2} = 0.5 \quad P(\bar{F}) = \frac{1}{2} = 0.5$$

$$P(E|F) = \frac{7}{9} = 0.777... \quad P(E|\bar{F}) = \frac{3}{7} = 0.428$$

$$P(F|E) = \frac{0.777 \cdot 0.5}{0.777 \cdot 0.5 + 0.428 \cdot 0.5} = 0.644$$

Какво щеше да се случи, ако имахме повече кутии F_j ?
Обобщена теорема на Бейс:

$$P(F_j|E) = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^n P(E|F_i)P(F_i)}$$

10.10 Филтри за спам

Множество (извадка) за обучение:

Лоши (спам) и-мейли: B

Нормални и-мейли: G

Един потребител класифицира всеки и-мейл в обучаващата извадка като нормален или лош (спам) и-мейл.

Откриване на думи, които се появяват в множествата B и G.

$n_B(w)$ Брой на спам мейлите с думата w

$n_G(w)$ Брой на нормалните мейли с думата w

$p(w) = \frac{n_B(w)}{|B|}$ Вероятност един спам-мейл да съдържа думата w

$q(w) = \frac{n_G(w)}{|G|}$ Вероятност един нормален и-мейл да съдържа w

S : Събитие, при което и-мейла X е спам

E : Събитие, при което и-мейла X съдържа думата w

$$P(S|E) = \frac{P(E|S)P(S)}{P(E|S)P(S) + P(E|\bar{S})P(\bar{S})}$$

$$P(S) = \frac{1}{2} \quad P(\bar{S}) = \frac{1}{2} \quad P(E|S) = p(w) \quad P(E|\bar{S}) = q(w)$$

Ако $P(S|E) > 0.9$ имейла се отхвърля.

Пример 10.10.1. Обучаваща извадка за думата "Ролекс":

"Ролекс" се появява в 250 от 2000 спам и-мейли.

"Ролекс" се появява в 5 от 1000 нормални и-мейли.

Ако един нов и-мейл съдържа думата "Ролекс" каква е вероятността този и-мейл да е спам?

$$n_B(Rolux) = 250$$

$$n_G(Rolux) = 5$$

$$p(w) = \frac{250}{2000} = 0.125$$

$$q(w) = \frac{5}{1000} = 0.005$$

S : Събитие, при което и-мейла X е спам

E : Събитие, при което и-мейла X съдържа думата Rolex

$$P(S|E) = \frac{P(E|S)P(S)}{P(E|S)P(S) + P(E|\bar{S})P(\bar{S})}$$

$$P(S) = \frac{1}{2} \quad P(\bar{S}) = \frac{1}{2} \quad P(E|S) = 0.125 \quad P(E|\bar{S}) = 0.005$$

$$P(S|E) = \frac{0.125 \cdot 0.5}{0.125 \cdot 0.5 + 0.005 \cdot 0.5} = 0.961 > 0.9$$

По-добрите филтри за спам използват едновременно две думи

$$P(S|E_1 \cap E_2) = \frac{P(E_1|S)P(E_2|S)}{P(E_1|S)P(E_2|S) + P(E_1|\bar{S})P(E_2|\bar{S})}$$

11 Лекция 10: Теория на алгоритмите

11.1 Алгоритми

Дефиниция 11.1.1. *Алгоритъмът е крайно множество от точно формулирани операции (инструкции) за реализация на някакво изчисление или решаване на даден проблем.*

Свойства на алгоритмите:

- Входни данни -от дадено множество
- Изходни резултати -от дадено множество (решение)
- Яснота на всяка стъпка от изчислениет
- Коректностна изходните резултати за всички възможни входни данни
- Краен брой стъпкипри реализацията на изчислителнията
- Дадена сложностна всяка изчислителна стъпка
- Общовалидностза клас от задачи.

11.2 Сложност

Нека като пример разгледаме два алгоритъма А и В, които решат един и същ клас задачи.

Сложността по време на А е $5000n$, а тази на В - $\lceil 1.1n \rceil$ за входна поредица от n елемента.

При $n = 10$, А се нуждае от (изпълнява) 50000 стъпки, а В само от 3.

Изглежда че В е по-добър от А.

При $n = 1000$, обаче, А се нуждае от 5 000 000 стъпки, докато В има нужда от $2,5 \cdot 10^{41}$ стъпки. Това означава, че В не може да се използва за големи входни последователности, докато А продължава да бъде приложим.

Възниква въпросът кой от алгоритмите е по-сложен?

В случая за оценка на сложността на алгоритмите е въведена мярка. Тази мярка обикновено е някаква функция, която зависи от размерността на входните данни.

Оказва се, че това което е важно е ръстана функцията на сложността (сложността) с нарастване на размерността на данните.

Ръста на сложността по време и "пространство" (обем на изходните данни) с увеличаване на размера на входната последователност е подходяща мярка за сравнение на алгоритмите.

Алгоритмичната сложност за дадено изчисление е най-общо мярка за това колко трудно е да се извърши изчислението.

Тоест, тя измерва някои страни на цената на дадено изчисление (в общия смисъл на "цена").

Количеството ресурси необходими за изчислението:

Някои от най-основните мерки за сложност са:

Сложност по време: брой необходими операции (стъпки)

Сложност по "пространство": броя необходими битове памет

Сложността обикновено се изразява като функция надължината на входната поредица (обема на входните данни).

11.3 Ръст на функция

Да допуснем, че алгоритъмът А има сложност по време при най-лош случай (worst-case time complexity - w.c.t.c или само време) $f(n)$ за вход с дължина n , докато алгоритъмът В (за същата задача) има време $g(n)$. Кой алгоритъм ще бъде по-бърз при всички достатъчно големи входни поредици от данни – т.нар. най-тежък случай (worst-case)?

Ръстът на дадена функция обикновено се записва със операцията голямо-О.

Дефиниция 11.3.1. Нека f и g са функции на цели или реални числа и резултатите от тях са реални числа.

Казваме че, $f(x)$ е $O(g(x))$ ако съществуват константите C и k , такива че:

$$|f(x)| \leq |O(g(x))| \quad \forall x > k$$

Когато анализираме ръста на функцията на сложността, $f(x)$ и $g(x)$ са винаги положителни

Ето защо можем да опростим горното условие до

$$f(x) \leq C \cdot g(x) \quad \forall x > k$$

Ако искаме да покажем, че $f(x)$ е $O(g(x))$, трябва да намерим само една двойка (C, k) (която никога не е уникална).

Идеята, която стои зад операцията главно-О, е да се установи горна граница за ръста на функцията $f(x)$ за големи стойности на x .

Тази граница се определя от функцията $g(x)$, която обикновено е много по-проста от $f(x)$

Въвеждаме константата C в изискването $f(x) \leq C \cdot g(x)$ при всяко $x > k$, защото C не расте заедно с x .

Интересуват ни случаите за големи стойности на x , затова няма проблем ако $f(x) > C \cdot g(x)$ $x \leq k$.

Често използвани функции $g(n)$ за оценка, подредени от най-бавно към най-бързо нарастване :

- 1
- $\log n$
- n
- $n \log n$
- n^2
- n^3
- 2^n
- $n!$

Задача, за която сложността за най-тежкия случай е полином се нарича решима (възможно решима - tractable)

Задачи с по голяма сложност се наричат трудно решими (intractable)

Задачи, чието решаване не съществува алгоритъм се наричат нерешими (unsolvable)

Полезни правила за Главно - О

- $\forall f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0, \quad a_0, a_1, \dots, a_n \in \mathbb{R} \implies f(x) = O(x^n)$
- $f_1(x) = O(g_1(x)), f_2(x) = O(g_2(x)) \implies (f_1 + f_2)(x) = O(\max(g_1(x), g_2(x)))$
- $f_1(x) = O(g(x)), f_2(x) = O(g(x)) \implies (f_1 + f_2)(x) = O(g(x))$
- $f_1(x) = O(g_1(x)), f_2(x) = O(g_2(x)) \implies (f_1 f_2)(x) = O(g_1(x) g_2(x))$

11.4 Машина на Тюринг

Машината на Тюринг е въображаемо изчислително устройство, описано от английския математик Алън Тюринг през 1936 г.

Тюринг използва машината, за да даде първото точно определение на понятието алгоритъм (наричано още механична, формална или ефективна процедура). Машината се използва при доказването на основни резултати в компютърните науки, най-вече в областите изчислимост и сложност на алгоритмите, както и в математическата логика.

Машината на Тюринг се състои от четири компонента:

- Памет — безкрайна лента, състояща се от клетки, във всяка от които е записан символ от някаква крайна азбука. Азбуката съдържа специален празен символ (обикновено обозначаван с '0') и един или повече други символи. Във всеки момент от работата на машината лентата е крайна, но при нужда може да и залепяме отляво или отдясно нови клетки, съдържащи празния символ.
- Глава, която във всеки момент от изчислението се намира над определена клетка от лентата. При всеки такт главата прочита символа от клетката, над която се намира, записва нов символ и се премества наляво или надясно по лентата в зависимост от изпълняваната инструкция и прочетения символ.
- Програма — краен списък от инструкции, който за разлика от съвременните компютри е отделен от паметта. Всяка инструкция е поредица от указания какво да се направи, ако главата е прочела i -тата буква от азбуката. Всяко указание съдържа информация какъв символ да се запише обратно върху лентата, коя инструкция ще се изпълнява на следващата стъпка и накъде (наляво или надясно) да се премести главата.
- Регистър, съдържащ номера на активната в момента инструкция (програмен брояч). Една от инструкциите се приема за начална, т.е. изчислението започва със зареждането на номера ѝ в програмния брояч. Има и крайна инструкция — при достигането ѝ изчислението спира.

12 **Лекция 11: Математическа индукция**

13 Лекция 12: Рекурсия

13.1 Рекурсивно дефинирани редици

Често пъти е трудно членовете на даден обект или числова редица да бъдат явно изразени.

Например: Редицата на Фибоначи:

$$\{f_n\} = 0, 1, 1, 2, 3, 5, 8, 13, \dots \Leftrightarrow f_n = f_{n-1} + f_{n-2} \quad f_0 = 0, f_1 = 1, n > 1$$

Може, обаче, да съществува някаква "локална" връзка, която да доведе до рекурсивна дефиниция-формула, която изразява по-горните членове на дадена редица чрез по-долните.

13.2 Рекурсивни дефиниции и Индукция

Рекурсивните дефиниции и индукционните доказателства взаимно се допълват: рекурсивните дефиниции обикновено позволяват да бъдат доказани по естествен начин зависимости, отнасящи се за рекурсивно дефинирани редици.

Това следва от формата на рекурсивната дефиниция, състоящ се от две части:

Инициализация - аналогична на базисните случаи при индукция

Рекурсия - аналогична на индукционната стъпка

Както при индукцията, така и при рекурсията аналогията с домино е изключително полезна.

13.3 Рекурсивни функции

Можем да разглеждаме всяка функция с дефиниционно множество \mathbb{N} като числова редица и обратно.

Пример: $f_n = f(n)$

Например, редицата на Фибоначи става функция на Фибоначи, както следва:

$$f(0) = 0 \quad f(1) = 1 \quad f(2) = 1 \quad f(3) = 2 \dots$$

Такива функции след това могат да се дефинират рекурсивно, като се използва дефиницията за рекурсивна редица.

За да се намери дадена комбинация от n и k се гледа къде се пресичат n -тия ред и k -тата диагонална колона

Рекурсивна формула

$$C(n, k) = \begin{cases} 0, & k < 0 \text{ или } k > n \\ 1, & k = n \\ C(n-1, k-1) + C(n-1, k), & \text{в останлите случаи} \end{cases}$$

13.4 Рекурсивни дефиниции

13.4.1 Функция най-голям общ делител (Евклид)

$$\gcd(x, y) = \begin{cases} x, & y = 0 \\ \gcd(y, x \bmod y), & \text{в останалите случаи} \end{cases} \quad (x > 0)$$

13.4.2 Математически операции

- Събиране

$$m + n = \begin{cases} m, & n = 0 \\ m + 1, & n = 1 \\ (m + (n - 1)) + 1, & n > 1 \end{cases}$$

- Сума

$$\sum_{i=1}^n a_i = \begin{cases} 0, & n = 0 \\ \sum_{i=1}^n a_i + a_n, & n > 0 \end{cases}$$

- Произведение

$$\prod_{i=1}^n a_i = \begin{cases} 1, & n = 0 \\ (\prod_{i=1}^n a_i) \cdot a_n, & n > 0 \end{cases}$$

13.4.3 Множества

Понякога множества могат да се дефинират рекурсивно. Започва се с базово множество елементи, и рекурсивно чрез определени операции се дефинират все повече елементи. В такъв случай множеството включва всички елементи, които могат да бъдат получени от базовото множество чрез краен брой позволени операции.

14 **Лекция 13: Крайни автомати**

Разглеждаме система за управление на една врата, отваряща се само в една посока.

Системата има две състояния:

Отворено

Затворено

Системата има два входа:

Човек, намиращ се в позиция А

Човек, намиращ се в позиция В

Правила за управление на вратата:

Ако вратата е затворена, тя трябва да се отвори само, ако човека се намира в А, а не в В.

Вратата трябва да се затвори, ако няма хора в А и В.

14.1 **Крайна последователност от действия**

Даден краен автомат (КА) обикновено се представя като ориентиран граф:

Ориентирания граф се състои от два обекта:

Състояния (наричани още възли или върхове)

Ребра със стрелки, които представляват позволените преходи.

Едно от състоянията обикновено се приема за начално (начален връх).

Ние ще се разгледаме един упростен модел на т. нар. крайни разпознаватели, при които някои от състоянията са избрани за крайни състояния.

Един краен автомат чете редица от символи (от входна азбука), наречена входна дума, и извежда също редица от символи (от изходна азбука), наречена изходна дума.

В зависимост от програмата си, крайният автомат притежава определен краен брой състояния, в които може да се намира. В началото се намира в едно специално състояние, наречено начално състояние.

Работата на един краен автомат се състои от определен брой стъпки, като на всяка стъпка той чете следващия символ на входната дума. В зависимост от прочетения символ и състоянието, в което се намира, автоматът извежда редица от изходни символи и преминава в ново състояние.

14.2 Азбуки и стрингове

Стрингъте последователност от символи.

Множеството от символи, от които е формиран даден стринг се вземат от дадено крайно множество наречено азбука.

Стринг без елементи се нарича празен стринг и се бележи с ϵ . Ако A е азбука, безкрайното множество от всички стрингове формирани от A се бележи с A^* .

14.3 Езици

Езикът е множество от стрингове.

Ако A е азбука, то език формиран от A е множеството от стрингове, чиито компоненти са от множеството A .

По този начин A^* е най - големия възможен език формиран от A , и всеки друг език формиран от A е под множество на A^* .

14.4 Регулярни езици

Има няколко начина за описание на регулярните езици:

Езици, които могат да бъдат разпознати от някой краен автомат

Езици, които могат да бъдат формирани индуктивно чрез комбинации от много прости езици.

Езици, които се описват с регулярни изрази.

Всеки език, породен от специална (много ограничена) граматика.

Започва се от много проста база от езици и се формират по-сложни езици, като се комбинират езиците от базата по различни начини:

База: $\emptyset, \{\epsilon\}, \{a\}$ са регулярни езици $\forall a \in A$.

Индукция: Ако L и M са регулярни езици, то следните езици също са регулярни:

$$L \cup M \quad L \circ M \quad L^*$$

14.5 Операция *

Това е унарна операция, върху множества от стрингове, или множества от символи.

Ако V е множество от стрингове, V^* е най-малкото множество, което включва V , съдържа ϵ (празния стринг) и е затворено по отношение

на операцията конкатенация (свързване на стрингове). Това множество, също може да бъде описано, като се използва операцията конкатенация на нула или повече стрингове от V .

Ако V е множество от символи то V^* е множеството от всички стрингове, които се получават от символите от V , като се включва и празния стринг.

14.6 Автомати за работа със стрингове

Входните данни могат да се представят чрез стрингове посредством някаква азбука и това определя как автоматът преминава от едно състояние в друго.

Започвайки от началното състояние, символите на входния стринг водят до промяна на състоянието на автомата.

Крайните разпознаватели дават отговори само "да" или "не" според това дали се намират в крайно състояние

Стринговете, които водят до крайно състояние се приемат (разпознават) от автомата.

Разрешените крайни състояние са означени с две окръжности. Често може да има повече от едно такова състояние.

14.7 Детерминиран краен автомат (ДКА)

При детерминирани съществува само една възможност за прехода в ново състояние, докато при недетерминирани са възможни няколко различни прехода.

Когато разчитаме символите на стринга, се местим от едно състояние в друго. Някои от стринговете ще доведат автомата до разрешено крайно състояние, а други не.

ДКА, използващ азбука с краен брой символи, представлява краен насочен граф, като от всеки възел излиза по един клон за всеки символ от азбуката A .

За такъв автомат знаем еднозначно в кое състояние се намира при зададен входен стринг.

ДКА приема стринг w от A^* , ако съществува път от началното до някое крайно състояние, така че w е съвокупността от символите по пътя от началния до крайния връх.

В противен случай ДКА отхвърля w .

Съвкупността от всички стрингове, които се приемат от ДКА M се нарича език на автомата M и се означава с $L(M)$.

14.8 Функция на преходите

ДКА може да бъде представен също така и чрез функция на преходите, която се означаваме с T и се записва така : $T(i,a) = j$.

За пълно описание на ДКА трябва да знаем:

какви са състоянията;

кое е началното и кое е крайното състояние;

множеството от преходите между състоянията.

14.9 Недетерминиран краен автомат (НКА)

ДКА се наричат детерминирани, защото следвайки даден входен стринг, знаем точно в кое състояние се намира автомата и начина за достигането му.

При НКА понякога има повече от една възможна посока в графа при зададен входен символ.

Неопределеност възниква, защото при следването на определен стринг, автоматът може да заеме едно от няколко възможни състояния или да стигне до дадено състояние по различни пътища! Недетерминиран краен автомат (НКА), използващ азбука A , се представя чрез краен насочен граф като от всеки възел излизат 0 или повече ребра:

Всяко ребро се означава с буква от A или с \backslash (неопределеност).

Няколко ребра с еднакво означение могат да излизат от един и същи възел (неопределеност).

Възможно е да няма асоциирано ребро за всяка буква от азбуката. Стрингове, които съдържат символи без асоциирано ребро за тях се отхвърлят (не могат да бъдат разпознати).